Николайчук Я. М.

КОДИ ПОЛЯ ГАЛУА:

теорія та застосування

Монографія

Тернопіль
2012

УДК 511.2; 681.215
ББК 65.050.9(2)я73
Н 63
Рецензенти:

Задірака В. К., доктор фізико-математичних наук, професор, член-кореспондент Національної академії наук України, завідувач відділу Інституту кібернетики ім. В. М. Глушкова НАН України

Мельник А. О., доктор технічних наук, професор, завідувач кафедри „Електронні обчислювальні машини" Національного університету „Львівська політехніка"

Дивак М. П., доктор технічних наук, професор, декан факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету

Н 63 **Николайчук Я.М.**

Коди поля Галуа: теорія та застосування./Монографія/ - Тернопіль: ТзОВ "Терно-граф", 2012. - 392 с., іл.

ISBN 978-966-654-233-8

У монографії викладено фундаментальні основи теорії чисел у різних теоретико-числових базисах. Вперше зроблена спроба узагальнення теорії, методології, моделей і алгоритмів формування, перетворення, опрацювання, передавання, а також побудови спецпроцесорів у кодах поля Галуа. Сформовані базові принципи формалізованого опису широкого класу абелевих груп та полів Галуа. Особливу увагу приділено теорії та технології формування, перетворення, опрацювання та передавання інформаційних потоків на основі кодів поля Галуа в розподілених комп'ютерних системах реального часу. Викладена теорія генерування кодів поля Галуа на основі незвідних поліномів виконання арифметичних операцій у кодах Галуа та побудови відповідних спецпроцесорів на різних рівнях розподілених комп'ютерних систем. Розглянуто процеси вдосконалення компонентів комп'ютеризованих систем на основі кодів поля Галуа, а також їх реалізації в комп'ютерних системах. Викладена теорія кодових шкал та дисків в базисі Галуа. Розглянуті теоретичні підходи та методи нелінійного кодування джерел інформації на основі адаптивних методів, інтегрально-імпульсної технології у теоретико-числовому базисі Галуа та процедур рандомізації цифрових повідомлень. Узагальнена теорія перетворення та опрацювання цифрової інформації на основі багатоканальних кодів Галуа в середовищі теоретико-числового базису Крестенсона. Розроблені теоретичні засади міжбазисних перетворень Радемахера-Галуа та Радемахера-Крестенсона. Приділено увагу теоретичним основам побудови спецпроцесорів кореляційного опрацювання інформації в різних теоретико-числових базисах. Подані структурні та схемотехнічні рішення формувачів, перетворювачів, засобів реєстрації та передавання інформації, а також спецпроцесорів у теоретико-числовому базисі Галуа, які застосовані в спеціалізованих та проблемно-орієнтованих комп'ютерних системах у різних галузях промисловості та виробництва. Монографія призначена для науковців і спеціалістів у галузі "Інформатика" та "Комп'ютерні технології" і може бути корисна для аспірантів та студентів, які навчаються за напрямками "Технічна кібернетика","Комп'ютерна інженерія", "Комп'ютерні науки" та "Системна інженерія".

ББК 32.973.2

ISBN 978-966-654-233-8

СТИСЛИЙ ЗМІСТ

ЗМІСТ

СПИСОК СКОРОЧЕНЬ

G-технологія – інформаційна технологія на основі кодів Галуа;
АЛП – арифметико-логічний пристрій;
АЦК – аналого-цифровий кодер;
АЦП – аналого-цифровий перетворювач;
БСТП – багатопараметричний сенсор температурних полів;
БМ – базовий модуль;
Д – дані;
ДІ – джерело інформації;
ДІКМ – диференціально-імпульсна кодова модуляція;
ДФСЗК – досконала форма системи залишкових класів;
ЕОМ – електронно-обчислювальна машина;
ЗІ – захист інформації;
ІІТ – інтегральна інформаційна технологія;
ІМ – інформаційні моделі;
ІТ – інформаційна технологія;
ККД – коефіцієнт корисної дії;
КМ – комп'ютерна мережа;
КМ – кластерна модель;
КНМ – контролер низової мережі;
КС – комп'ютерна система;
КФ – кореляційна функція;
КТЗ – Китайська теорема про залишки;
КПГ – коди поля Галуа;
ЛГ – логічний граф;
ЛСІМ – логіко-статистична інформаційна модель;
МДІ – модель джерела інформації;
НСД – найбільший спільний дільник;
О – оператор;
ОЗП- оперативно запам'ятовуючий пристрій;
ОК – об'єкт керування;
ОУ – об'єкт управління;
ОС – обчислювальна система;
ПЗ – програмне забезпечення;
ПЗП – постійний запам'ятовуючий пристрій;
ПІ – приймач інформації;
ПКД – пам'ять колективного доступу;
ПЛІС – програмовані логічні інтегральні схеми;
Р – процесор;
РКС – розподілена комп'ютерна система;
САПР – система автоматизованого проектування;

СД – структуризовані дані;
СЗК – система залишкових класів;
СКС – спеціалізовані комп’ютерні системи;
СО – системний об’єкт;
СОІ – середовище обробки інформації;
СПД – система передавання даних;
СПІ – середовище передавання інформації;
ТЧБ – теоретико - числовий базис;
RSA – алгоритм Rivest-Shamir-Adleman;

Присвячується 200-річчю
від Дня народження
великого математика
<u>Евариста Галуа</u>

ВЕЛИКИЙ ГАЛУА І ЗАСТОСУВАННЯ ЙОГО ТЕОРІЇ У НАШ ЧАС

Більше двохсот років відділяє нас від того часу, коли народився Еваріст Галуа – видатний французький математик, заслуги якого визнані вченими всього світу. Його життя обірвалося дуже рано – у 20 років. Тепер Галуа – гордість французької науки, кращі риси якої відображені у його роботах, що майже на століття випередили розвиток фундаментальної математики.

Еваріст Галуа народився 25 жовтня 1811 року в Бур-ля-Рен - передмісті Парижа. У віці 12 років вступив у королівський коледж Луі-де-Граж. У 16 років Е. Галуа почав читати фундаментальні математичні твори, у тому числі мемуари Нільса Абеля про розв'язок рівнянь довільного степеня, де довів, що для рівнянь степеня 5 і вище розв'язки "в радикалах" неможливі. Ця тема захопила Евариста, він почав власні дослідження і просунувся у теорії набагато далі. Галуа знайшов необхідну та достатню умову для того, щоб корені рівняння виражалися в радикалах. Найціннішим був не тільки цей результат, а й ті методи, за допомогою яких Галуа вдалося його отримати.

Уже в 17 років Е. Галуа опублікував свою першу наукову роботу в журналі "Annales de Gerdonne". У 1829 році він вступає у Вищу нормальну школу, в якій провчився тільки один рік і був виключений за підтримку республіканців.

Але талант не допомагав визнанню Е. Галуа, тому що його рішення перевищували рівень професійності викладачів, виясненню його розумових та математичних висновків не сприяло також те, що він не старався чітко викладати їх на папері й часто пропускав очевидні для нього положення. На

думку математиків, саме математика перетворила його зі слухняного учня у видатного вченого. Правда, не всі математики того часу відповідно оцінили його наукові праці.

Наприклад, про роботу Галуа в двох частинах схвально відгукнувся відомий математик Коші, якому вона була відправлена на рецензію. Цей рукопис був загублений Коші і не потрапив у Паризьку Академію на конкурс математичних робіт.

Негаразди у науковому житті Е. Галуа тривають. Мемуар про свої відкриття Е. Галуа посилає Фур'є для участі у конкурсі на приз Паризької Академії, але через кілька днів Фур'є раптово помирає, не встигнувши опрацювати цей текст. Приз Паризької Академії отримує Нільс Генрік Абель.

Стаття, відіслана Сімеону-Дені Пуассону, була повернута з наступною резолюцією: "…У всякому випадку, ми зробили все від нас залежне, щоб зрозуміти доведення пана Галуа. Його міркування та виклади не володіють ні достатньою ясністю, ні достатньою повнотою для того, щоб ми могли бути впевнені та вважати їх точними, тому ми не в змозі дати про них уявлення в цій доповіді".

Чому саме ім'я Галуа і його теорія стали предметом широкомасштабних фундаментальних досліджень великої когорти вчених усього світу? Тому, що саме життя і творча діяльність Евариста Галуа - приклад альтернативи класичній теорії математики - нескінченно малих приростів аргументів та функцій для інтегрального і диференціального числення. Ця теорія, яка два століття тому стала альтернативою, альтернативно розвивається у наш час.

У книзі Андре Дальма є коротка характеристика діяльності Евариста Галуа: "…погано миряться з думкою, що геній може приєднатися до прогресивного руху народу. Щоб мати право відрізнятися від інших, учений, перш за все, повинен дати докази своєї безпечності. Якщо він не є цілком безпечний, то добиваються, щоб він став таким".

За спогадами вчителів Королівського коледжу, Е. Галуа був "не вживчивий, дивний і надмірно балакучий", мав "неабиякі здібності" та "досить нестандартні манери".

При житті Галуа не добився ніякої слави, хоча сам ніколи не сумнівався в цьому. Сучасні на той час математики не тільки не розуміли, що його роботи знаменують собою нову епоху в розвитку математики, але навіть не звернули на них уваги. Лише більш, ніж через півстоліття науковий світ оцінив оригінальність і глибину його мислення, а також властивий Галуа дар передбачення.

У наш час знаменитими стали не тільки спеціальні наукові проблеми, але й окремі прогнози Галуа, в яких передбачені нова система організації науки і комунікації вчених, котрими, як правило, завжди нехтували.

Від самого початку Галуа відмовлявся від шкільних підручників, автори яких мистецтво мислити підміняли мистецтвом вводити в оману за допомогою наборів певних слів. Поглинувши книгу Лажандра “Елементи геометрії” і роботи Лагранжа “Рішення численних рівнянь”, “Теорія аналітичних функцій”, “Лекції з теорії функцій” Галуа прийшов до ідеї і фундаментального поняття групи. Ознайомившись із роботами Ейлера, Гаусса і Якобі, Галуа висловив здивування методами розв’язків задач, якими користувалися викладачі. А викладачем математики був 33-хрічний професор Рішар, в якого у цей час вчився знаменитий математик Шарль Ерміт, котрий зберіг для людства рукописи робіт Галуа. Рішар мав велике задоволення відкривати таланти. Розв’язки математичних задач, які пропонував Галуа, захоплювали його. У записах Рішар зазначив: “Галуа працює тільки в вищих основах математики”.

Шістдесят написаних рукою Е. Галуа сторінок відкрили світові ім’я цього воістину геніального вченого. Галуа цікавили насамперед не окремі математичні задачі, а загальні ідеї, що визначають увесь ланцюг усвідомлень і скеровують хід думок. Його доведення базовані на глибокій теорії, яка дає змогу об’єднати всі досягнення математики і визначити розвиток науки надовго вперед. Гільберт назвав цю теорію “визначенням певної архітектури понять симетрії”.

Отже, за 20 років життя Е. Галуа встиг зробити відкриття, що прирівнюють його до найвидатніших математиків XIX століття. Розв’язуючи задачі з теорії алгебричних рівнянь, він заклав основи сучасної алгебри, вийшов на такі фундаментальні поняття, як група (Е. Галуа перший використав цей термін, досліджуючи симетричні групи) і поле (тепер кінцеві поля названі полями Галуа).

Відкриття Е. Галуа започаткували новий напрям – теорію абстрактних алгебричних структур. У наступні 20 років Келі та Жордан розвивали й узагальнювали ідеї Галуа, які якісно перетворили теоретичне обличчя всієї математики.

Огюст Шевальє, товариш Е. Галуа, і молодший брат Е. Галуа Альфред послали останні роботи Галуа Гаусу і Якобі, але відповіді не отримали. Тільки у 1843 році відкриття Галуа зацікавили Ліувілля, який опублікував і прокоментував їх.

Роботи Галуа з теорії груп ознаменували кінець передісторії і початок істинної історії математики.

“Я займаюсь аналізом аналізу. Я хочу підкорити обчислення своїй волі, згрупувати математичні операції, навчитись їх класифікувати за степенем трудності, а не за зовнішніми прикметами – це завдання математиків майбутнього”, - пише Галуа.

Поняття групи визначається як сукупність абстракцій, що мають певні спільні властивості. Наприклад група цілих чисел, група простих чисел, група невід'ємних залишків тощо.

Абстракціями в групах можуть бути операції. Саме цей випадок вивчив Галуа. Суттєво, що ні самі абстракції, ні операції над ними ніяк не конкретизовані. Вивчаючи властивості та користуючись формальними правилами перетворення груп, можна передбачити їхні властивості, а значить, у принципі, вирішити конкретну задачу.

Тут проявляється альтернатива мислення Галуа, про що Паскаль сказав: "Дійсний дослідник відкриває передовсім не нові об'єкти, а нові зв'язки між ними, які визначають можливі властивості відомих і невідомих об'єктів".

Таким чином, геніальне прозріння Галуа полягає у розробленні однієї з найфундаментальніших альтернативних теорій математики, головне значення яких полягає у пізнанні того, що ідея груп симетрій, яка була виключно пов'язана з геометрією, грає фундаментальну роль у всій математиці й взагалі у природознавстві. У тому числі групою Галуа класичної механіки є група Галілея, а механіки теорії відносності – група Лоренца. Класичними стали відомі групи Абеля й абстрактна теорія груп О. Ю. Шмідтта.

А як протікає розвиток ідей Галуа у наш час? Кажуть, що немає нічого практичнішого, ніж хороша теорія.

Проте в школах теорію полів Галуа навіть не згадують, у технічних вузах – глибоко не вивчають. На математичних факультетах університетів теорію груп вивчають у загальному розділі теорії чисел. Останній раз праці Галуа були видані російською мовою в 1935 році й перевидані у 2003 році.

Я тривалий час вивчав різні теоретико-числові базиси та системи числення, в т.ч. мінус двійкову та систему залишкових класів, яка базується на теорії полів Галуа і намагався боротись із недоліками двійкової системи, яка з легкої руки Фон-Неймана і К. Шеннона зайняла провідне місце у розвитку комп'ютерних систем та цифрових мереж передавання інформації.

Потрібно було об'єднувати канали, стискати повідомлення, захищати дані від завад у лініях зв'язку і помилок в обчислювальних машинах. Потрібний був якийсь універсальний, гнучкий та ефективний спосіб кодування чисел, повідомлень і сигналів на всіх стадіях руху інформації від джерел даних, об'єктів управління – до процесорів опрацювання даних та серверів, їх компактного зберігання на носіях і в базах знань.

Двійкова система числення не могла розв'язати цю проблему в такому розумінні, бо виявилася частковим випадком полімодульної системи залишкових класів.

Після кількох лекцій професора І. Я. Акушського з теорії залишкових класів, які я прослухав у 1970 році у Санкт-Петербурзі в Інституті точної

механіки і оптики, я внутрішньо крикнув “еврика” і детально простудіював книжку І. Я. Акушского і Д. І. Юдіцкого “Машинна арифметика в залишкових класах”. Згодом – книгу академіка В. М. Амербаєва “Основи машинної арифметики комплексних чисел”, В. М. Торгашева “Коректуючі коди в системі залишкових класів”, Б. М. Коляди “Теорія і застосування математичних машин”.

Система числення залишкових класів, так звана СЗК, базована на тому, що в кожному розряді число має окремий модуль, тобто свій набір цифр – залишків, ансамбль яких по простому модулю якраз утворює групу у полі цілих чисел, яке є одночасно полем Галуа.

Фундаментальна математична основа СЗК базована на доведенні кількатисячолітньої давності так званої Китайської теореми про залишки, математичною основою якої є властивості операцій над ідеалами.

У наш час числення залишкових класів знайшло ряд надзвичайно ефективних застосувань при побудові спеціалізованих супершвидкодіючих обчислювальних машин, наприклад, до 30-ти мільярдів операцій на секунду, з надвисокою надійністю і нечутливістю до відмов обладнання та помилок обчислень.

Є ряд аргументів, які теоретично й експериментально дослідив професор А. Канн, в котрого я прослухав курс лекцій з біоніки, що в пучках нейронів використовується кодування імпульсних сигналів у базисі груп, полів Галуа та залишкових класів.

Застосування теорії груп привело до сучасних успіхів у розробці швидкодіючих алгоритмів і процесорів цифрових перетворень Фур’є на основі чисел Мерсена, Ферма та Фібоначчі.

Я детально досліджував теорію канонічних форм прямого і зворотного перетворення залишкових класів. У тому числі: цілочисельне, нормалізоване і знайдене мною досконале перетворення СЗК.

На базі цих перетворень розроблені та впроваджені у техніку ефективні методи і технічні засоби стискування та надійного передавання цифрової інформації на низових рівнях комп’ютеризованих систем управління.

У той самий час числення залишкових класів виявилося безпосереднім чином зв’язане з перетвореннями комплексних чисел у базисах Фур’є, Гільберта та Ейлера. Це числення виявилося частковим випадком фундаментальної теорії систем базисних функцій, які саме й утворюють найдосконаліші групи.

У порядку наростання загальності довелося вивчити базисні функції Радемахера, Хаара, Уолша, Віленкіна-Крестенсона.

Стало зрозумілим, що базис Радемахера – це прямокутні симетричні функції, які породжують двійкову систему числення. Функції Хаара породжують розрядно позиційні системи числення. Функції Уолша

породжують відомі коди Грея і Хеммінга. Вибірка функцій Уолша, для яких властива операція рекурсії, породжує двійкові коди і числення Галуа. Сюди належать популярні коректуючі коди Голея, Боуза-Чоудхурі-Хоквінгема та інші, що базовані на діадних операціях зсуву. Цим базисом породжуються так звані шумоподібні сигнали (М-сигнали) та коди Баркера, які знайшли винятково необхідне застосування у космічній галузі, на наддалекому підводному зв'язку. На принципах застосування кодів Баркера побудовані сучасні сотові комп'ютерні радіомережі та системи передавання повідомлень у каналах з інтенсивними завадами.

Числення залишкових класів виявилося досконалим, частковим випадком функцій Віленкіна-Крестенсона, що склали основу базисних перетворень Галуа.

Така фундаментальна генеалогія числення базису Галуа віщувала надію надзвичайно широкого застосування у різних галузях сучасної науки – молекулярній хімії, механіці, обчислювальній техніці, кібернетиці, теорії сигналів, теорії інформації, молекулярній біології, соціології, екології, енергетиці та інших галузях знань.

Числення Галуа виявилося справді плідним ельдорадо на сучасному етапі розвитку кібернетичної науки і техніки. Важливою ідеологічною властивістю числення Галуа є нове ефективне рішення задач, які стандартно вважаються неймовірними та супроводжуються психологічними бар'єрами і дуже часто недовірливими запитаннями й константаціями типу: «Це бути не може тому, що воно не може бути».

Розглянемо ряд ефективних прикладів застосування числення Галуа при розв'язуванні задач у галузі автоматики і цифрових обчислювальних систем, що становлять галузь моєї наукової діяльності.

Перше – це кодові шкали Галуа. В наш час розроблені однобітові нереверсивні, двохбітові реверсивні та трьохбітові позиційні змінно-якісні кодові шкали Галуа, що відкривають нові широкі можливості побудови високоточних вимірювачів переміщень, кута повороту і позиціювання транспортних засобів в одно-, дво- і трьохмірних декартових, а також полярних координатах. Особливістю таких шкал є незалежність числа кодових доріжок від довжини кодової шкали.

Друге – це аналого-цифрові перетворювачі Галуа паралельно-послідовного скануючого типу. Унікальною властивістю таких АЦП є висока регулярність обчислювального середовища, дворазове зменшення числа компараторів стосовно до наявних структур і практичне виродження дешифратора, що визначає значну перспективу їх реалізації на базі великих інтегральних схем.

Третє – це формування місцевого, регіонального або всесвітнього Галуа-часу. Галуа-час - це генерація в ефір кратносекундних бітів Галуа, сприймаючи які, можна синхронізувати всі електронні годинники, створити

зелені хвилі світлофорів у містах, керувати виробничими машинами, технологічними процесами та персональним обладнанням.

Четверте – це алгоритми і процесори стискування інформації, які порівняно з адаптивним кодуванням за ефективністю переважають останні у два-три рази. Основу таких алгоритмів становить рекурентна зв'язність елементів поля Галуа, що дає змогу нумерувати активні цифрові відліки одним або двома бітами Галуа незалежно від довжини послідовності та числа пропущених неактивних цифрових повідомлень.

П'яте – це побудова електронної пам'яті з колективним паралельним доступом на базі адресації Галуа, що дозволяє багатьом віддаленим абонентам із нешвидкодіючим обладнанням одночасно звертатися до бази даних і одночасно отримувати файли даних нешвидкісними каналами зв'язку. Галузь застосування – інформаційно-пошукові системи, бази знань, довідкові системи в бібліографії, медицині та ін.

Шосте – це побудова інтелектуальних сенсорів і перетворювачів енергетичних параметрів з інтегрально-імпульсним кодом Галуа. На основі таких перетворювачів нині впроваджені необхідні комерційні комп'ютерні системи контролю та обліку енергоспоживання цехів і промислових підприємств.

Сьоме – це розвиток теорії обчислень у полі Галуа, побудова нового покоління процесорів, обчислювальних систем та низових обчислювальних мереж на базі сформованої нами вертикальної інформаційної технології в базисі дискретних перетворень Галуа.

Цю нумерацію можна було б продовжити. Але найвизначнішим може бути застосування ідей Галуа до розкриття фундаментальних теоретичних основ побудови кодів генетичної основи всього живого – коду молекули ДНК.

Справді, якщо позначити букви коду ДНК A, T, G, C – відповідно 0, 1, 2, 3, то ми одержимо не що інше, як послідовність Галуа за модулем чотири. Ми вже генеруємо за допомогою комп'ютерів синтезовані ДНК-подібні послідовності. На стадії розроблення теорія спірально-циліндричних полів Галуа. Наприклад: у полі Галуа немає повторень і ДНК не має повторень. Код Галуа має липкі кінці і ДНК має липкі кінці. У полі Галуа будь-яку помилку можна виявити за допомогою ключів і в ДНК рибосома зупиняється в процесі синтезу білка, коли виявляє недопустиму помилку. Код Галуа можна розмежувати, наприклад, на межі букв AT і потім абсолютно точно відновити попередню послідовність. Подібно, в ДНК рестріктаза розриває біологічну структуру на границі конкретних букв, а лігаза абсолютно точно відновлює попередню структуру, яка спроможна після цього правильно створити організм.

Кожна амінокислота кодується одним або кількома триплетами, причому всі триплети – кодони, число яких рівне Шеннонівському числу

4^3=64. Деякі з кодонів використовуються тільки як розділові знаки, а промотори – як мітки блокової синхронізації. ДНК – це яскравий приклад досконалої групи симетрій, і вона не може мати іншу теоретичну основу, крім теорії Галуа.

Автор висловлює подяку академіку НАН України, професору О. В. Палагіну – фундатору теорії розподілених комп'ютерних систем за ідею інтегрованого підходу до теорії джерел інформації та міжбазисних перетворень, рецензентам: члену-кореспонденту НАН України, професору В. К. Задірaці, професору А.О.Мельнику та професору М. П. Диваку за цінні рекомендації, науковим співробітникам Карпатського державного центру інформаційних засобів і технологій Технічного центру НАН України, директору Інституту мікропроцесорних систем керування об'єктами електроенергетики, доценту І.О.Сабадашу, директору Інституту проблемно-орієнтованих комп'ютерних систем, доценту М.І.Чирці, які очолюють названі наукові установи Карпатського центру НАН України, працівникам кафедри спеціалізованих комп'ютерних систем факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету за участь в обговоренні та допомогу в оформленні книги.

Щиро дякую своєму науковому вчителеві заслуженому діячеві науки і техніки д.т.н, професору А.М.Лучуку за те багатство ідей та щедрий науковими результатами шлях, яким він мене направив.

Особливу подяку висловлюю моїй дружині Любові Николайчук, кандидату юридичних наук, доценту кафедри соціальних комунікацій та права Івано-Франківського національного технічного університету нафти і газу за підтримку і творчу співпрацю.

Монографія орієнтована на фундаментальні засади кібернетики, теорії та методології інформатики, інформаційної теорії ДНК, розподілених комп'ютерних систем та комп'ютерної криптології, викладені в наукових працях академіків НАН України В.С.Дейнеки, І. В. Сергієнка, О. В. Палагіна, член-кореспондентів НАН України Б.М.Малиновського, В.П.Боюна, В. К. Задіраки, А.М.Гупала, професорів С.Г.Буніна, А.О.Мельника та інших вчених і системно охоплює результати сумісних досліджень учнів наукової школи автора: доктора технічних наук, професора Л. Б. Петришина; кандидатів технічних наук С. М. Іщерякова, Г.Я.Ширмовського, С. І. Мельничука, В. В. Яцківа, А. І. Сегіна, Н. Г. Яцків, І. М. Лазаровича, Н.Д.Круцкевича, О. М. Заставного, І.Р.Пітуха, Н. Я. Возної, Т. М. Гринчишина, кандидата фізико-математичних наук М. М. Касянчука та аспірантів І. З. Якименка, І. О. Погонця, І.Б. Албанського, О. І. Волинського, П. В. Гуменного, Т.О.Заведюк, А. Р. Воронича, В. В. Шаряка, С. В. Івасьєва.

РОЗДІЛ 1

ЗАГАЛЬНІ ПИТАННЯ ТЕОРІЇ ЧИСЕЛ

1.1. Множини та операції.

1.1.1. Бінарні алгебраїчні операції.

Відношення між елементами множини називаються алгебраїчними операціями. До основних алгебраїчних операцій слід віднести: додавання і множення чисел, многочленів, алгебраїчних дробів, додавання векторів площини, кон'юнкція та диз'юнкція висловлень, додавання і множення функцій, композиція відповідностей. Дані операції виконуються над парами елементів однієї й тієї самої множини, тобто над парами чисел, многочленів, векторів, функцій і називають їх бінарними алгебраїчними операціями, або бінарними операціями. Загальне означення бінарної алгебраїчної операції, якому задовольняють, зокрема, перелічені вище операції, формулюють так.

Означення. Нехай M – довільна множини елементів *a, b, c, ...* Бінарна операція в множині M – це закон, за яким будь-яким двом елементам a і b цієї множини, взятим у повному порядку, ставиться у відповідність єдиний елемент цієї множини.

З означення бінарної операції випливає, що входить вимога однозначності операції і її здійсненності для будь-яких двох елементів множини, та впорядкованість елементів множини M при виконанні над ними операції. Це означає, що парам елементів a, b і b, a ставляться у відповідність, взагалі кажучи, різні елементи множини M.

Загальне означення бінарної операції говорить, що віднімання цілих, раціональних чисел, додавання векторів, розміщених у деякій площині $a, є$ бінарні операції. До бінарних операцій слід також віднести диз'юнкцію й кон'юнкцію висловлень, додавання й множення дійсних функцій від змінної x, визначених для всіх дійсних значень x, композиція (множення) відображень деякої множини A в себе. Знаходження найбільшого спільного дільника і найменшого спільного кратного двох натуральних чисел – бінарні операції в множині натуральних чисел.

Слід зазначити, що наведене означення бінарної операції є досить широке, проте воно не охоплює всіх математичних операцій. Наприклад, утворення скалярного добутку двох векторів площини a не є бінарною операцією в множині M всіх векторів цієї площини, бо скалярний добуток двох векторів є число, а не вектор, і, отже, не є елементом множини M. Так само операція знаходження спільного дільника натуральних чисел m і n не є бінарною операцією в множині натуральних чисел, бо хоч вона здійсненна

завжди, але не є однозначною: числа m і n можуть мати кілька спільних дільників.

Для скорочення запису кожну конкретну операцію позначають спеціальним знаком: додавання позначають знаком «+», множення – знаком «•», операцію перетину двох множин – знаком «∩» і т. д. При вивчені загальних властивості бінарних операцій, які притаманні багатьом конкретним бінарним операціям, то говорять не про конкретні, а про довільні операції. Для позначення довільних бінарних операцій вводиться символи τ *і* $*$ – елемент, якому бінарна операція τ (операція $*$) ставить у відповідність пари елементів a і b, позначатимемо символом а τ b $(a * b)$ і називається *композицією* елементів a і b, елементи a і b – *членами* композиції.

Якщо композиція $a \tau b(a * b)$ дорівнює елементу c, то це відношення записується $a \tau b = c$ $(a * b = C)$ і читається «композиція a і b дає c». Якщо над елементами множини виконується одна чи декілька бінарних операцій багатократно, то порядок їх виконання, робиться як над операціями з числами, вказується за допомогою дужок.

Отже, будь-яка бінарна операція τ в множині M є деяке відображення:

$$\varphi: M \times M \to M. \qquad (1.1)$$

Бінарні алгебраїчні операції можна також розглядати і як тернарні відношення. Нехай у множині M задано бінарну операцію $*$. Це означає, що будь-якій упорядкованій парі (a, b) елементів $a \in M, b \in M$ ставиться у відповідність єдиний елемент $a * b = c \in M$. Нехай символ P_* це сукупність усіх упорядкованих трійок (a, b, c) елементів з M таких, що $c = a * b$. Очевидно, що $P_* \subset M^3$ і тому $\rho_* = (P_*, M)$ – тернарне відношення між елементами множини M, при цьому $\rho_* abc = [a * b = c]$. Отже, бінарна операція $*$ рівносильна тернарному відношенню ρ_*.

1.1.2. Властивості бінарних операцій.

Нехай τ – бінарна операція, визначена в множині M.

Означення. Бінарна операція τ називається асоціативною, якщо $\overset{\forall}{a,b,c} \in M$ $[(a \tau b) \tau c = a \tau (b \tau c)]$, а якщо в множині M є принаймні одна трійка елементів $a,b,c \in M$ така, що $(p \tau b) \tau c \neq a \tau (b \tau c)$, то така операція називається неасоціативною.

Множина цілих чисел з операціями додавання і множення, як відомо, асоціативні. Асоціативні також операції об'єднання і перетину підмножин

даної множини, а операція віднімання не асоціативна, бо *(17 - 5) - 3 ≠17 – (5 - 3).*

Означення. Бінарна операція τ комутативна, якщо $a,b \overset{\forall}{\in} M$ *[a τ b = b τ a],* і якщо в множині *M* є принаймні $a,b \in M$ одна пара елементів *a* і *b* таких, що *a τ b ≠b τ a,* то вона некомутативна.

Операції об'єднання і перетину підмножин множини А, комутативні. Комутативні також операції додавання й множення цілих чисел. Операція ж віднімання цілих чисел некомутативна, бо *23 - 7 ≠ 7 - 23.*

Припустимо, що у множині *M* визначені бінарні операції *τ* і * .

Означення 3. Бінарна операція * є дистрибутивна відносно операції τ, якщо:

$$a,b,c \overset{\forall}{\in} M \; [(a\tau b) * c = (a * c)\tau(b * c) \; i \; c * (a \tau b) = (c * a) \tau (c * b)]^1. \quad (1.2)$$

Операція * не дистрибутивна відносно операції *τ*, якщо в множині *M* є принаймні одна трійка елементів *a, b* і *c*, для яких не справджується жодне з записаних вище рівностей.

Операція множення цілих чисел дистрибутивна відносно операції додавання їх, але операція додавання не дистрибутивна відносно операції множення, оскільки *7+4·5≠(7+4)·(7+5).* З'ясуємо тепер значення асоціативності й комутативності бінарних операцій.

Роль комутативності дає можливість переставляти місцями елементи, до яких застосовується бінарна операція, при цьому спрощуються формули й міркування.

Асоціативність дає можливість означити композицію трьох і взагалі будь-якого скінченного числа елементів, взятих у певному порядку.

Приклад. Нехай дано три елементи *a, b i c* множини *M*. Вирази виду *a τ b τ c* поки що не визначені, невідомо, як слід розуміти під композицією цих трьох елементів, оскільки в означенні бінарної операції говориться про композицію лише двох елементів, узятих в певному порядку. Однак вирази *(a τ b)τ c і a τ (b τ c)*– це композиція елемента *a τ b* і елемента c, другий – композиція елемента *a* і елемента *b τ c*. Оскільки *τ* асоціативне, то обидві ці композиції дорівнюють одному і тому самому елементу множини *M*.

Даний елемент приймається за композицію *a τ b τ c*, записану вже без дужок. Таким чином, рівність *a τ b τ c =(a τ b) τ c = aτ(bτ c)* можна розглядати як означення композиції *a τ b τ c* трьох елементів *a, b, c,* узятих в тому порядку, в якому вони записані.

---

[1] Якщо операція * комутативна, то в означенні дистрибутивності операції * говорять про справедливість лише однієї з рівностей (a τ b) * c = (a * c) τ (b * c) і c * (aτ b) = (c * a) τ (c * b), оскільки в цьому випадку кожна з цих рівностей є наслідком іншої.

Якщо дано n елементів множини M, записаних у певному порядку: $a_1, a_2 \dots a_n$. Для даних елементів можна кількома способами розставити дужки, що вказують порядок виконання бінарної операції τ над цими елементами. З даного твердження слідує наступна теорема.

Теорема. Результат послідовного виконання асоціативної операції над елементами упорядкованої множини $a_1, a_2 \dots a_n$ порядку, вказаному за допомогою дужок, не залежить від способу розставляння дужок, тобто при різних розставляннях дужок результати будуть рівні між собою.

Доведення. Ця теорема доводиться методом математичної індукції. Для n=3 теорема справедлива. Нехай теорема справедлива для будь-якого натурального числа k, меншого ніж n і не меншого ніж 3, тобто що результат послідовного виконання операції над елементами упорядкованої множини $a_{l1}, a_{l2} \dots a_{lk}$ визначений однозначно. Доведемо, що в такому разі теорема справедлива і для n. Для цього насамперед треба показати, що для кожного натурального числа $k < n-1$ справджується рівність:

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_k \tau a_{k+1}) \tau (a_{k+2} \tau a_{k+3} \tau \dots \tau a_n)$.

Це справді так. Композиції $a_1 \tau a_2 \tau \tau \dots a_k$ і $a_{k+2} \tau a_{k+3} \tau \dots \tau a_n$, за припущенням, однозначно визначені. Нехай:

$a_1 \tau a_2 \tau \dots \tau a_k = b$, $a_{k+2} \tau a_{k+3} \tau \dots \tau a_n = c$.

Тоді:

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = b \tau (a_{k+1} \tau c)$,

$(a_1 \tau a_2 \tau \dots \tau a_k \tau a_{k+1}) \tau (a_{k+2} \tau a_{k+3} \tau \dots \tau a_n) = (b \tau a_{k+1}) \tau c$.

Із асоціативності операції τ

$b \tau (a_{k+1} \tau c) = (b \tau a_{k+1}) \tau c$

і, отже,

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_k \tau a_{k+1}) \tau (a_{k+2} \tau a_{k+3} \tau \dots \tau a_n)$

Із справедливості початкової рівності випливає, що для будь-яких натуральних чисел k і l, менших ніж n, справджується рівність:

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_l) \tau (a_{l+1} \tau a_{l+2} \tau \dots \tau a_n)$.

Не втрачаючи загальності міркувань, можна вважати, що $l > k$. Нехай $l = k + s$. Тоді

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_k \tau a_{k+1}) \tau (a_{k+2} \tau a_{k+3} \tau \dots \tau a_n) =$

$= (a_1 \tau a_2 \tau \dots \tau a_{k+1} \tau a_{k+2}) \tau (a_{k+3} \tau a_{k+4} \tau \dots \tau a_n) = \dots = (a_1 \tau a_2 \tau \dots \tau a_{k+s-1}) \tau (a_{k+s} \tau a_{k+s+1} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_l) \tau (a_{l+1} \tau a_{l+2} \tau \dots \tau a_n)$

і, отже,

$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_l) \tau (a_{l+1} \tau a_{l+2} \tau \dots \tau a_n)$

Припустимо тепер, що в системі елементів $a_1, a_2 \dots a_n$ двома будь-якими різними способами розставлено дужки, що вказують, в якому саме порядку треба послідовно виконувати операцію τ. Треба показати, що в обох випадках результат виконання операції буде той самий.

Справді, якщо послідовно виконувати операцію τ в порядку, що вказується розставлянням дужок першим способом, то останнім кроком буде виконання операції τ над композиціями:

$$a_1 \tau a_2 \tau \dots \tau a_k \text{ і } a_{k+1} \tau a_{k+2} \tau \dots \tau a_n \ (1 \le l \le n-1).$$

При виконанні операції τ в порядку, вказаному розставлянням дужок другим способом, останнім кроком буде виконання операції τ над деякими композиціями:

$$a_1 \tau a_2 \tau \dots \tau a_l \text{ і } a_{l+1} \tau a_{l+2} \tau \dots \tau a_n \ (1 \le l \le n-1).$$

Але з попередніх рівностей:

$$(a_1 \tau a_2 \tau \dots \tau a_k) \tau (a_{k+1} \tau a_{k+2} \tau \dots \tau a_n) = (a_1 \tau a_2 \tau \dots \tau a_l) \tau (a_{l+1} \tau a_{l+2} \tau \dots \tau a_n).$$

Отже, в обох випадках матимемо той самий результат. Тому, за принципом математичної індукції, теорема справедлива для будь-якого натурального $n \ge 3$. Цим теорему доведено.

Якщо в композиції $a_1 \tau a_2 \ \tau \dots \tau a_n$ всі члени дорівнюють одному й тому самому елементу a, то позначатимемо її символом $\overset{n}{\tau} a$. Застосовуючи дану теорему до композиції, що містить однакові члени, тоді:

$$\overset{m+n}{\tau} a = (\overset{m}{\tau} a) \tau (\overset{n}{\tau} a), \qquad \overset{mn}{\tau} a = \overset{m}{\tau} (\overset{n}{\tau} a),$$

де m і n – будь-які натуральні числа.

Якщо бінарна операція τ, яка визначена в множині M, не лише асоціативна, а й комутативна, тоді справджується наступне твердження:

Твердження. Композиція будь-яких n елементів $a_1, a_2 \dots a_n$ множини M не залежить від того, в якому порядку йдуть її члени.

В алгебрі вивчають бінарні операції, які за своїми властивостями більш-менш близькі до операцій додавання і множення чисел, і тому кожну з них також називають або додаванням, або множенням. Якщо бінарну операцію, визначену в множині M, називають додаванням, то тоді елемент c, який цією операцією ставиться у відповідність упорядкованій парі елементів $<a, b>$, називають сумою цих елементів, а елементи a і b – доданками і записують: $a + b = c$; якщо ж її називають множенням, то c називають *добутком* елементів a і b, а a і b – співмножниками і записують: $a \cdot b = c$, або $ab = c$.

Означення. Суму n доданків, кожен з яких дорівнює елементу a, називають n-кратним елемента a і позначають символом na.

Означення. Добуток n співмножників, кожен з яких дорівнює елементу a,

називають n-м степенем елемента a і позначають символом a^n.

Формули (1.3) в символах операцій додавання і множення записуються відповідно:

$$(m + n)a = ma + na,\ (mn)\ a = m\ (na) \qquad (1.3)$$

і

$$a^{m+n} = a^m\ a^n,\ a^{mn} = (a^n)^m.$$

Отже, використання властивостей основних бінарних операцій дозволяє спростити обрахунки та зменшити складність операцій.

1.1.3. Обернені операції. Нейтральний елемент, симетричні елементи.

Нехай у множині M визначена бінарна операція τ.

Означення 6. Для визначеної бінарної операції τ у множині M здійсненна обернена бінарна операція, тоді і тільки тоді, коли для будь-яких елементів a і b множини M існує одна і тільки одна пара таких елементів $x°$ і $y°$, що $a\ \tau\ x° = b$ і $y°\ \tau\ a = b$. Крім цього, якщо операція τ комутативна, то елементи $x°$ і $y°$ однакові.

Якщо τ є комутативна операція додавання (множення), то обернену їй операцію $\perp$ називають відніманням (діленням). Елемент $x°$, що задовольняє умови $a\ \tau\ x° = x°\ \tau\ a = b$, називають різницею (часткою) елементів b і a і записують x°= b-a (x°= b: a, або $x° = a / b$).

Обернена операція $\perp$, очевидно, не є новою незалежною операцією: вона є похідною від операції τ.

Означення 7. Елемент $\eta\ \in M$ називається нейтральним елементом відносно операції τ, якщо $\overset{\forall}{a \in M}\left[a \perp \eta = a \wedge \eta \tau a = a\right]$. Так, у множині всіх підмножин деякої множини M порожня підмножина Ø є нейтральним елементом відносно операції об'єднання U, а M – відносно операції перетину $\cap$. Число 0 є нейтральним елементом відносно додавання цілих чисел, а число 1 – відносно їх множення. Нейтральним елементом відносно композиції (множення) відображень множини M в M є тотожне відображення M на M.

Нейтральний елемент відносно операції додавання, визначеної в деякій множині M, називають нульовим елементом і позначають символом 0, а відносно операції множення – одиничним елементом (одиницею) і позначають символом e. Неважко довести

Теорему. Якщо в множині M з бінарною операцією τ в нейтральний елемент η, то тільки один.

Означення. Нехай у множині M з бінарного операцією τ є нейтральний елемент η. Елемент a' називають симетричним елементу $a\ \in M$, якщо $a'\ \tau\ a = a\ \tau\ a' = \eta$.

Нейтральний елемент ŋ, очевидно, симетричний сам собі. Якщо a – відмінне від нуля раціональне число, то число – a симетричне йому відносно додавання, $\frac{1}{a}$ число відносно множення.

Елемент множини M, симетричний елементу $a \in M$ відносно операції додавання, називають протилежним a і позначають символом $-a$, а симетричний відносно операції множення називають оберненим a і позначають символом a^{-1}.

Теорема. Якщо бінарна операція τ, визначена в множині M, асоціативна, то для будь-якого елемента а множини M в ній може існувати не більше, ніж один симетричний елемент.

Доведення. Справді, якщо a' і a'' – елементи, симетричні елементу a, то

$a' = a' \tau \eta = a' \tau (a \tau a'') = (a' \tau a) \tau a''=\eta \tau a''= a''$, тобто $a' = a''$.

Справді, якщо a' і a'' – елементи, симетричні елементу a, то $a' = a' \tau \eta = a' \tau (a \tau a'') = (a' \tau a) \tau a''=\eta \tau a''= a''$, тобто $a' = a''$.

Використання бінарних операцій та їх властивостей дозволяє зменшувати кількість операцій в складних задачах, а з подальшим врахуванням математичних основ теоретико-числових базисів їх складність суттєво зменшується.

1.2. Теоретико-числові базиси.

1.2.1. Математичні основи теоретико-числових базисів.

Фундаментальною основою теоретико-числових базисів (ТЧБ) є теорія чисел, вища алгебра та теорія функцій комплексного змінного.

Значний вклад у розвиток математичних основ ТЧБ та їх практичного застосування для створення технічних засобів цифрового опрацювання сигналів (ЦОС) внесли І.Я. Акушський, Л.В. Варіченко, М.Г Карповський, В.Г. Лабунець, А.М Трахтман, Р.Г. Фараджев, Н. Ахмед і Н. Рао, Р. Аргевал, Б. Гоулд і У. Рейдер, П. Рабинер, Р.Блейхут та інші.

М.А. Раков обґрунтував необхідність та перспективність ґрунтовного використання властивостей ґрунтовного використання цифрового подання сигналів та побудови моделей їх опрацювання.

Особливо застосування потужного математичного апарату Теорії чисел, системи Діофантових рівнянь, полів Галуа, залишкових класів та груп Абеля.

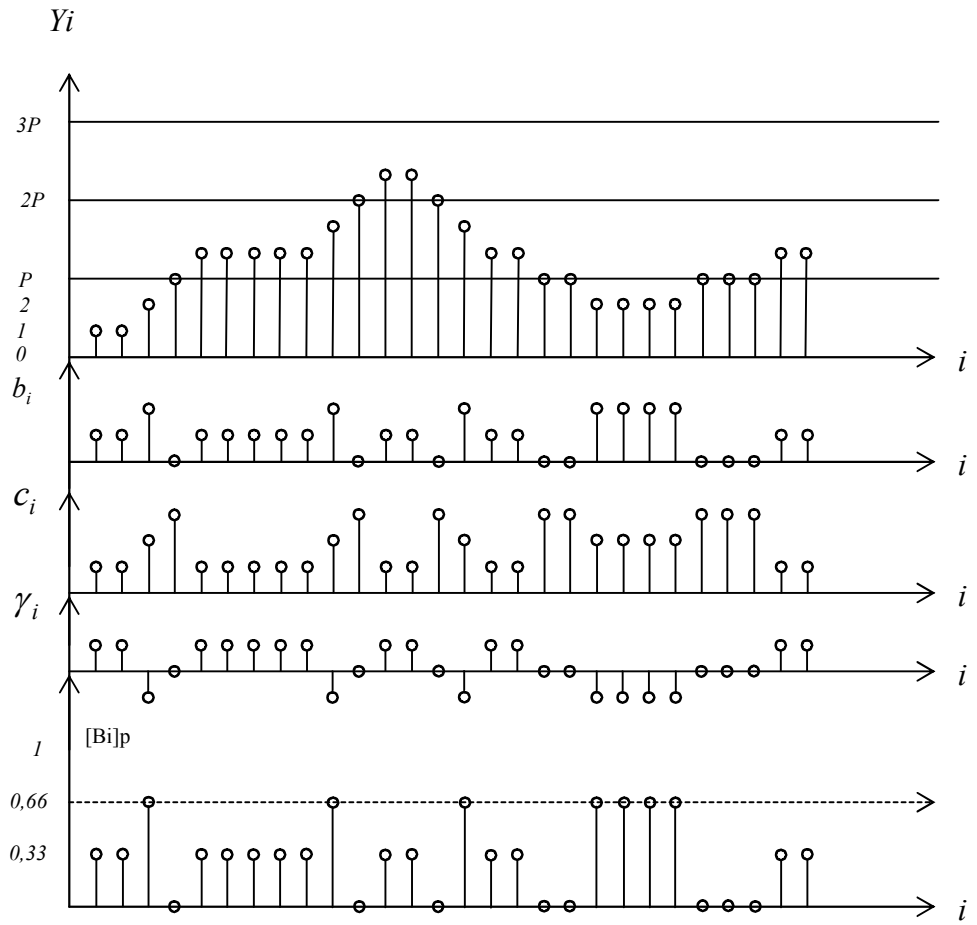


Рис.1.1. Кодування решітчатої функції залишками по модулю у базисі Галуа (P=3).

Найбільш загальним поняттям ТЧБ є алгебраїчна система, яка визначається властивостями Абелевої групи.

Важливо, що Абелева група комутативна $x * y = y * x$ відносно операцій (*) додавання та множення.

Число Абелевих груп рівне числу довільних розкладів числа в суму невідємних доданків.

Додавання елементів у Абелевій групі виконується по модулю $Z_i = x_i \otimes y_i (\mathrm{mod}\, P)$, P – ціле число.

Поля та кільця є алгебра з двома бінарними операціями. Прикладом кілець є поля цілих Z, раціональних Q, дійсних R, комплексних C, та гіперкомплексних G чисел.

Множина значень дискретних відліків сигналу, представлено решітчастою функцією у Хемінговому просторі є підмножиною кільця цілих чисел або поля комплексних чисел.

Важливим типом кілець, які породжують ТЧБ, коди та системи числення є кільця *F(x)* поліномів від однієї змінної з коефіцієнтами з деякого поля.

$$F(x) = a_0 + a, x + a_2 x^2 + ...a_m x^m = \sum_{i=0}^{m} a_i x^i ,$$

де a_i – коефіцієнти полінома, які належать полю Р.

Глибока подібність властивостей кілець Z та *F(x)* дозволяють будувати алгебраїчні структури в одному кільці та використовувати їх рішення в іншому. Виходячи з кільця Z, можна адекватно побудувати кільця класів залишків по модулю Р та кінцеві поля Галуа *GF(P)*.

Наприклад повна система залишків по модулю простого числа Р утворює просте поле Галуа.

У кільці *GF(P)[x]* існують незвідні поліноми будь якої степені. Тому можна отримати різні розширення поля *GF(P)*. Одним з популярних розширень що застосовуються у практиці цифрового опрацювання сигналів є складене поле Галуа $GF(P^n)$, n – ціле. Це поле належить до класу гіперкомплексних полів.

Складене поле $GF(P^n)$, володіючи подібним порядком, як і поле $GF(P)$ при суттєво менших значеннях Р забезпечує паралельне виконання операцій над компонентами багаточлена, який є елементом поля $GF(P^n)$.

В теорії та практиці формування перетворення та цифрового опрацювання інформаційних потоків отримали кільця залишків по модулю цілого числа *Zp*.

Поняття “залишок” відноситься до теорії чисел і визначається лінійною формою $y = ap + b,$ яка відповідає порівнянню $y \equiv b(\mathrm{mod}\, P)$ *або* $b = res\, y(\mathrm{mod}\, P),$ де *y,a,b,p* – цілі числа *a* – член числа у по модулю Р, $a = \breve{E}\left[\dfrac{y}{p}\right],$ де $\breve{E}[\cdot],$ – цілочисельна функція з округленням до меншого цілого; b – найменший невід’ємний залишок $a \le b \le p-1;$ $\equiv$ mod, res – відповідно символи порівняння, модуля та операції добування залишку.

Число залишків рівне Р та рівне числу класів по mod*P*.

Існує чотири типи залишків (рис.1.1) b – найменші невід’ємні $0,1,2,...,P-1;$ c – найменші додатні $1,2,...,P-1,P;$ γ – найменші (по

абсолютному значенню) $-(P-1)/2,...,-1,0,1,2,...,(P-1)/2$; $[b]_p$ – нормовані по модулю $0/P \le [b]_p \le (P-1)/P:$, $[b]_p = b/P.$

У базисах класифікованих залишків можливо побудувати чотири еквівалентні теоретичні системи, що відрізняються складністю перетворень та технічною реалізацією процесів.

Наприклад: $Y = 24_{(10)}, P = 5,$ тоді

$b = res24(\mathrm{mod}5) = 4;$

$c = res24(\mathrm{mod}5) + 1 = 5$

$\gamma = b - P = 4 - 5 = -1;$

$[b]_p = 4/5 = 0{,}8.$

Якщо число У представляється у 2-й степені числення, тобто $Y = 11001_{(2)}$, то нормалізований залишок $[b]_p$ записується у нормалізованій формі двійкового числа з фіксованою комою.

$[b]_p = [4]_5 = 0{,}11001100...$

При цьому операція додавання по модулю двох залишків в нормалізованій формі виконується по “mod 1” , тобто шляхом відкидання цілої частини отриманої суми числа $[b]_b$ записаних у 2-й системі числення з фіксованою комою.

Наприклад: $b_1 = 4; b_2 = 3; P = 5,$, отже

$b_0 = (b_1 + b_2) \bmod P = (4+3)(\mathrm{mod}5) = 2.$

Аналогічно: $[b_1]_5 = 0{,}110011_{(2)j} [b_2]_5 = 0{,}100110_{(2)},$

Звідси $[b_1]_5 = 0{,}110011$

$[b_1]_5 = \dfrac{0{,}110011}{1{,}011001} \ (\mathrm{mod}1) = 0{,}11001_{(2)} = b_0.$

Відновлення залишку модульної операції додавання нормалізованих значень $[b_i]_5$ виконується згідно виразу $b_j = \hat{E}([b_0]P),$, тобто

$b_j = \hat{E}(0{,}011001*101) = 1{,}111101_{(2)} = 2$

Виконання операції піднесення до квадрату залишку b_i по модулю P виконується згідно виразу

$b_{i+1} = res\, b_i^2 (\mathrm{mod}\, P)$

і потребує у цілочисельній формі послідовності виконання наступних операцій :

1) множення $b_i b_i = b_i^2;$

2) ділення b_i^2; на P;

3) виділення залишку від ділення b_{i+1} шляхом відкидання цілої частини (рангу) у результаті не ділення $a, b_{i+1} \Rightarrow b_{i+1}$;

У нормалізованій формі ці операції спрощуються і виконуються згідно виразу (1.4)

$$b_{i+1} = \widehat{E}(res[b_i^2]_p \bmod 1) \qquad (1.4)$$

наступним чином :

1) Множення $b_i[1]_p = [b_i]_p$, де $[1]_p = b_i / P$;

2) Множення $b_i[1]_p = [b_{i+1}]_p$;

3) Множення з округленням до більшого цілого $\hat{E}(P[b_{i+1}]_p) = b_{i+1}$

Наприклад: $b_i = 5;\ P = 11,$ визначити $b_{i+1} = res5^2 (\bmod 11) = 3$.

1) 5*0,09…=0,45…=$[5]_{11}$;
2) 5*0,45…=0,27…=$[5^2]_{11}$;
3) Ê(11*0,45) = Ê(2,99…) = 3 .

Таким чином операція ділення b_i^2 на модуль P у нормалізованій формі замінюється операцією відкидання цілої частини від результату перемноження $b_i[b_i]_p$ Очевидно, що нормалізоване значення $[1_p]$, при відомому P, що має місце у задачах криптографії, може бути визначене наперед і вибиратися з таблиці.

Теорія залишків базису Галуа також охоплює кільце цілих комплексних чисел $Z = a + bi; i = \sqrt{-1}$.

Причому в комплексній області залишок відповідає аргументу комплексного числа, який також описується лінійною формою по модулю P= 2π, що слідує з формул Ейлера:

$$\omega = e^{-iZ} = \cos Z + i \sin Z,$$

де $Z = Z_p(\cos\gamma + i\sin\gamma)$, Z_p – модуль комплексного числа, γ – аргумент.

$a = \sqrt[r]{-1} \in Z_p$ і в полі комплексних чисел $\sqrt[r]{-1} = e^{-\frac{2\pi}{p}a}\ a$.

При цьому функції $\psi_a(t) = a^{at}$ утворюють ортогональний базис у просторі функцій, заданих на цілочисельному відрізку 0, T(a,p) -1. Базиси утворювані в такому випадку відрізняються тим, що дозволяють позбутися комплексних множень на числа виду $\exp\left(-i\frac{2\pi}{P}dt.\right)$. Вибираючи $a=2^k$, $k=1,2,...$ отримують найбільше спрощення реалізації базисних операцій програмно-апаратними засобами.

З метою забезпечення високої точності опрацювання інформації у базисах полів Галуа, необхідно вибирати великі значення простих модулів Р, що задовольняють Діофантовому рівнянню $2^q \equiv 1(2^q - 1)$, що приводить до арифметики по модулю $P = 2^k - 1$ і $P = 2^k + 1$.

Такими числами є відомі числа Мерсена $P = 2^q - 1$, де q – просте число, і Ферма $F_n = 2^{2^n} + 1$, де n –ціле число.

Числа Мерсена широко застосовуються у комплексних ТЧБ Фур'є-Галуа над полем $GF(P^2)$ з числом точок $N = 2^q$.

Окремим випадком такого класу ТЧБ є базис "Крестенсона-Галуа.

Застосування чисел Ферма в якості модулів ТЧБ відоме під назвою перетворення Рейдера. Відомі також сукупності простих чисел, які використовуються у цифрових перетворювачах Фур'є-Шевілла.

У загальному, перетворення на базі чисел Мерсена найбільш прості в технічній реалізації у порівнянні з іншими арифметиками.

Загальними обмеженнями описаних перетворень, основаних на властивостях простих та розширених полів Галуа є :

1) Обмежені функціональні можливості та реалізація тільки перетворень Фур'є;

2) Особливі обмеження по числу точок перетворення та точності обчислень.

ТЧБ на основі сум полів Галуа.

При опрацюванні багатомірних сигналів та інформаційних потоків використовують перетворення над зваженими та прямими сумами полів Галуа.

У першому випадку теоретичною основою перетворень є Китайська теорема про залишки, з якої слідує:

Якщо $P_1 P_2 ... P_n = P$ прості, то кільце Z_P класів залишків по модулю Р ізоморфне прямій сумі полів $GF(P_i)$, $Z_p \approx GF(P_1) + GF(P_2) + ... + GF(P_k)$.

Прямий та зворотній ізоморфізм таких зважених полів заданий відображенням:

$$\psi_1 : a \to (a_1, a_{2,}..., a_k),$$

де aZ_p; $a \equiv a_i (\mathrm{mod}\, P_i)$; i Є $1, k$;

$$\psi_1 : a \to (a_1, a_{2,}..., a_k) \to a,$$

де $a = M_1 M_1^{-1} a_1 + ... M_k M_k^{-1} Q_k$;

У другому випадку $P_1, P_2, ..., P_к$ – взаємно-прості , а всі $M_1^{-1} = 1$, $i \in 1, k$.

Викладені теоретичні засади утворення ТЧБ показують, що відомі базиси Крестернсона, Радемахера-Крестенсена та Крестензона-Галуа, які засновані на властивостях згаданої Китайської теореми про залишки. Своєю фундаментальною основою базуються на властивостях зважених та прямих сум полів Галуа.

Породжені такими ТЧБ системи числення та коди поля Галуа є фундаментальними теоретичними основами сучасної арифметики чисел. Особливо це стосується системи числення залишкових класів базису Крестенсона та кодових систем Галуа.

1.2.2. Теоретико-числові базиси на основі кусково-постійних ортогональних функцій.

До основних дискретних теоретико-числових базисів належать унітарні функції та коди, функції Хаара та розрядно-позиційні коди, дискретно-фазові функції та коди Лібова-Крейга, функції Радемахера та двійкові коди, функції Грея та коди Грея, функції Уолша, функції Галуа та кодові системи Галуа. Вибір кодової системи, базису або системи функцій залежить від задачі, властивостей інформаційного потоку, умов застосування даних та інш.

Зокрема, базисами для виконання дискретних ортогональних перетворень і дискретного подання одновимірного інформаційного потоку зі скінченною енергією, визначеного в просторі $L_2[a,b]$ на часовому інтервалі $T=[a,b]$, є повні ортонормовані системи функцій. Вейвлет-аналіз здійснюється на основі ортогональних або базисів Ріса. Повними ортонормованими системами функцій у просторі $L_2[a,b]$ є тригонометрична система та дискретні експоненціальні функції – як базис перетворення Фур'є, системи Уолша, Хаара, функції пилкоподібного базису, Віленкіна-Крестенсона та інші, проте актуальною залишається задача визначення галузей, способів і методів їх ефективного застосування, формування та дослідження інших базисів. Визначення особливостей та ефективності застосування різних систем функцій для виконання дискретних перетворень і аналізу інформаційних потоків зумовлює необхідність дослідження властивостей цих систем, наведених у наступних викладках.

1.2.2.1. Унітарний базис.

В якості вихідних у засобах перетворення форми інформації широкого застосування набули унітарні коди, розрядність бінарного подання слова яких відповідає повній шкалі квантування діапазону перетворення N. Здійснити перехід до ефективніших кодів із меншою

розрядністю дозволяє аналітичне подання унітарних кодів і встановлення функціональних залежностей з іншими кодами чи системами кодування.

Для подання унітарних кодів використовуються унітарні функції:

$$Uni(m,\theta,i) = sign(\sin(2^m \pi(\theta + i \cdot 2^{-n}))), \qquad (1.5)$$

де $m = 0, 1, \ldots, n+1$ – порядок набору системи функцій; $n = \log_2 N$; N – модуль цілочислових дискретних значень системи; $\theta = t/T$; $(0 \le \theta < 1)$ – нормований параметр часу; $T = 2\pi$; t – потокове значення часу; $0 \le t < 2\pi$; $i = 0,1,\ldots,2^{n-m+1} - 1$ – порядковий номер функції в наборі порядку m.

Набір нульового порядку $Uni(0,\theta,i)$ містить *2N* функцій (рис.1.2).

Властивості унітарних функцій:

1. Система з перших *N* унітарних функцій порядку *m* є лінійно незалежною, оскільки виконується достатня умова лінійної незалежності: ранг матриці *N* функцій дорівнює кількості функцій *N*. Наступні *N* функцій є лінійними комбінаціями *N* перших.

2. Унітарні функції не ортогональні, оскільки $\int_0^1 Uni(m,\theta,i)Uni(k,\theta,j)d\theta \neq 0$.

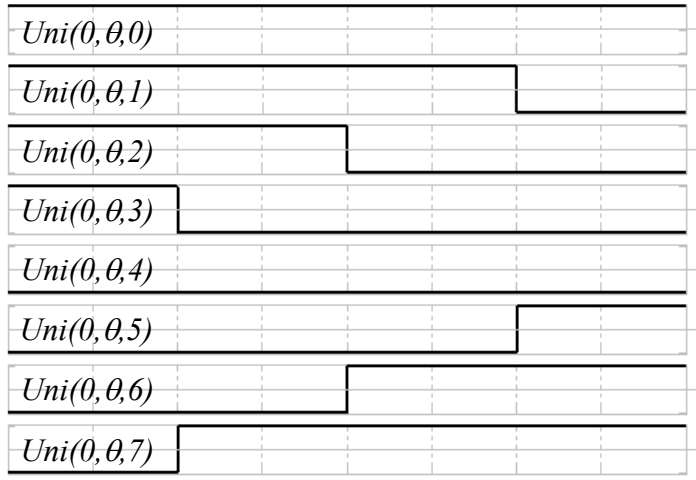


Рис.1.2. Унітарні функції нульового порядку.

Не ортогональність системи унітарних функцій зумовлює некомпактне пакування кодових елементів системи, що приводить до значної надлишковості інформаційних потоків. Внаслідок неортогональності та відсутності досліджень властивостей система не використовується як основа ТЧП.

Породжуючи кодову матрицю унітарного коду розміру $N \times N$ одержують при дискретизації з інтервалом $1/N$ за параметром часу перших $N=2^n$ із системи $2N$ унітарних функцій та здійсненні бінарної заміни значень функцій *1* на *0*, *–1* на *1* в точках $\theta_s = s/2^n$, $s=0,1,...,2^n-1$, яка реалізується за допомогою операції:

$$u_i = (1 - Uni(0, \theta_s, 2^n - 1 - i))/2, \qquad (1.6)$$

де $u_0, u_1, ..., u_i, ..., u_{2^n-1}$ – значення розрядів унітарного коду θ_s, $i=0,1,...,2^n-1$.

Для прикладу, при $n=3$ восьми функціям відповідають такі елементи кодової матриці:

$$\begin{array}{lcccccccc}
Uni(0,\theta,0) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Uni(0,\theta,1) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
Uni(0,\theta,2) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
Uni(0,\theta,3) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
Uni(0,\theta,4) & \rightarrow & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,5) & \rightarrow & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,6) & \rightarrow & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,7) & \rightarrow & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Наведені властивості системи унітарних функцій дозволяють у наступних викладках визначити процедури перетворення до інших базисів. У ролі первинних при перетворенні форми інформації та при переході від N-розрядних унітарних до кодів із меншою розрядністю також використовуються розрядно-позиційні коди.

1.2.2.2. Базис Хаара.

Основою розрядно-позиційних кодів є система функцій Хаара. Функції Хаара $Har(n,\theta,j)$ (рис.1.3) визначаються за формулою:

$$Har(n,\theta,j) = \begin{cases} 2^{\frac{n-1}{2}} sign(\sin 2^n \pi\theta), & \frac{j}{2^{n-1}} \le \theta < \frac{j+1}{2^{n-1}}, \\ 0 \ \textit{при інших} \ \theta \in [0,1), \end{cases} \qquad (1.7)$$

де $n = 0,1,..., \log_2 N$; $j = 0,1,...,2^{n-1}-1$; ($j=0$ при $n=0$), $0 \le \theta < 1$.

При реалізації ТЧП використовуються наступні властивості системи Хаара.

1. Функції Хаара $\{Har(n,\theta,j)\}$ утворюють повну ортонормовану систему в просторі інтегрованих із квадратом функцій $L_2[0,1)$, що дає можливість використовувати систему в якості базису для виконання ортогонального перетворення, яке є вейвлет-перетворенням.

2. На значній частині інтервалу визначення функції дорівнюють нулю, що дає можливість скоротити кількість арифметичних операцій при обчисленні перетворення. У результаті зменшується час обробки інформації.

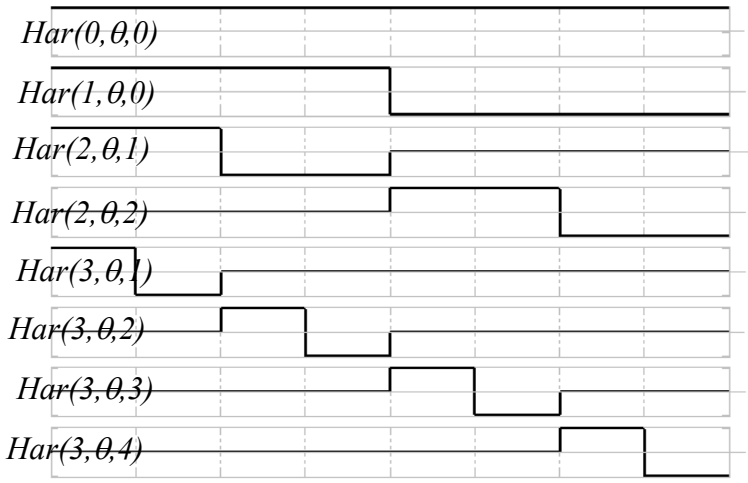


Рис.1.3. Система функцій Хаара.

Ненормований базис Хаара $\{sign(\sin 2^n \pi\theta)\}$ (без нормуючого множника $2^{\frac{n-1}{2}}$), в якому функції набувають значень ±1, 0, є основою розрядно-позиційних кодів. Розрядно-позиційні коди застосовуються в засобах перетворення форми інформації, в якості проміжних при аналогово-цифровому перетворенні, в давачах переміщень, для ініціювання комірок пам'яті тощо.

Для встановлення аналітичних співвідношень зв'язку систем функцій, які лежать в основі перетворення даних із розрядно-позиційного коду та в розрядно-позиційний код, використовуються розрядно-позиційні функції:

$$RP(i,\theta)=\begin{cases}-1, \dfrac{i}{2^n}\le\theta\le\dfrac{i+1}{2^n},\\ 1 \ \textit{при інших}\ \theta\in[0,1],\end{cases} \qquad (1.8)$$

$n=0,1,2,\ldots$, $i=0,1,\ldots,2^n-1$.

Дискретне подання 2^n розрядно-позиційних функцій та бінарна заміна значень функцій *1* на *0*, *–1* на *1*, яка подається за допомогою виразу:

$$p_i = (1 - RP(i,\theta_s))/2 = \frac{1}{2^{n/2}} Har(n+1,\theta_s,i-1) \qquad (1.9)$$

де p_i – значення розрядів розрядно-позиційного коду, породжує кодову матрицю:

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ \hdashline 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{array}$$

За умови простої реалізації перетворення інформації унітарному та розрядно-позиційному кодам властивий недолік необхідності використання повної *N*-розрядної шини кодового подання даних для кодування *N* дискретних повідомлень. Це зумовлює необхідність переходу до ефективніших методів кодування зі зменшеною розрядністю кодів до $n = \log_2 N$ в системах Радемахера та Грея.

В якості проміжних при переході до *n*-розрядних кодів використовуються коди Лібова-Крейга, які дозволяють у два рази зменшити розрядність коду та характеризуються властивістю абсолютного позиціонування.

1.2.2.3. Базис Лібова-Крейга.

Залежність унітарних кодів з іншими встановлюється за допомогою системи дискретно-фазових функцій, що є основою кодів Лібова-Крейга.

Дискретно-фазові функції порядку m подаються згідно наступного аналітичного виразу:

$$Dyf(m,\theta,i) = sign(\sin(2^m \pi(\theta + i \cdot 2^{-n}))), \qquad (1.10)$$

де $i = 0, 1, ..., 2^{n-m+1}-1$ – порядковий номер функції в наборі порядку m.

Графіки дискретно-фазових функцій першого порядку наведені на рис.1.4.

Властивості дискретно-фазових функцій:

1. Система з *N* дискретно-фазових функцій є лінійно залежною, оскільки частина функцій системи є лінійною комбінацією інших функцій системи:

$$Dyf(m,\theta,j+2^{n-m}) = -Dyf(m,\theta,j),$$

де $j = 0,1,\dots,2^{n-m}-1$.

Внаслідок лінійної залежності перші *N* функцій не утворюють повної системи.

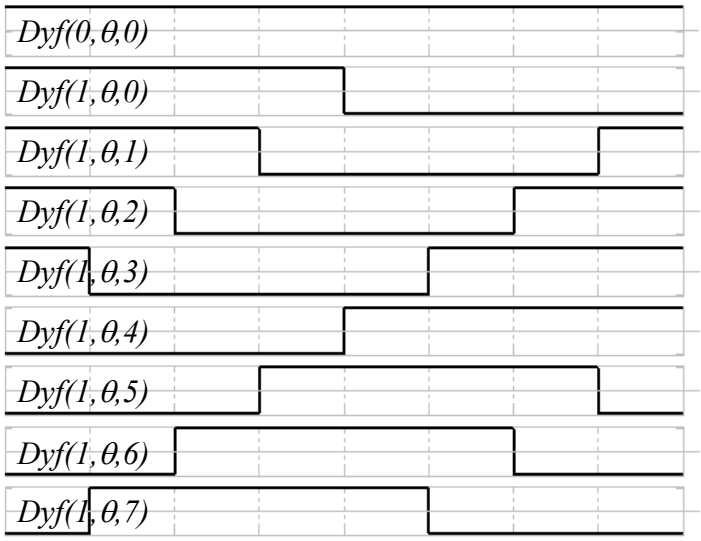


Рис.1.4. Дискретно-фазові функції першого порядку.

2. Система є неортогональною, тому що $\int_0^1 Dyf(m,\theta,i)Dyf(m,\theta,j)d\theta \neq 0$.

Породжуючи кодову матрицю розміру $\frac{N}{2}\times N$ одержують за допомогою дискретизації перших $N=2^{n-1}$ дискретно-фазових функцій порядку m та здійснення бінарної заміни значень функцій 1 на 0, –1 на 1, яка реалізується за формулою:

$$d_j = (1 - Dyf(1,\theta_s,2^{n-1}-1-j))/2, \qquad (1.11)$$

де $d_0, d_1, ..., d_j, ..., d_{2^{n-1}-1}$ – значення розрядів коду Лібова-Крейга $\theta_s = \frac{s}{2^n}$, $s=0,1,...,2^n-1$.

Наприклад, при N=8 елементи матриці відповідатимуть таким функціям

$$\begin{array}{rcccccccc} Dyf(1,\theta,0) & \to & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ Dyf(1,\theta,1) & \to & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ Dyf(1,\theta,2) & \to & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ Dyf(1,\theta,3) & \to & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ s & \to & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7. \end{array}$$

Внаслідок неповноти, система не використовується як основа для ортогональних ТЧП. Дискретно-фазові функції розглядаються як перехідні та як основа творення базисів і систем функцій Радемахера, Грея, кодування

даних в яких здійснюється з розрядністю $n = \log_2 N$ порівняно з N для унітарних кодів. Із цією метою в складі системи дискретно-фазових функцій виокремлюють дві підсистеми функцій виду $sign(\sin(2^n \pi\theta))$ та $sign(\cos(2^n \pi\theta))$.

1.2.2.4. Базис Радемахера.

Екстракція *sin*-складових набору дискретно-фазових функцій утворює систему функцій Радемахера (рис. 1.5).

$$Rad(n,\theta) = Dyf(n,\theta,0) = sign(\sin(2^n \pi\theta)) \qquad (1.12)$$

Рис.1.5. Функції Радемахера.

Система Радемахера є основою двійкової системи числення.

Відповідність між значеннями функцій у точках $\theta_s = s/2^n$, $s=0,1,...,2^n-1$ та їх поданням у двійковому коді $\theta_s = r_n r_{n-1} \ldots r_0$ встановлюється співвідношенням:

$$r_k = (1 - Rad(n-k,\theta_s))/2, \qquad (1.13)$$

де r_k – значення розрядів двійкового коду, $k = 0,1,...,n$.

Наприклад, при $n=3$ чотирьом функціям відповідають такі елементи кодової матриці розміру 4×8

$$\begin{array}{rcccccccc} Rad(0,\theta) & \to & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Rad(1,\theta) & \to & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ Rad(2,\theta) & \to & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ Rad(3,\theta) & \to & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ s & \to & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7. \end{array}$$

Наступні властивості системи Радемахера:

- функції Радемахера ортонормовані на відрізку $[0,1)$, оскільки:

$$\int_0^1 Rad(n,\theta)Rad(k,\theta)d\theta = 0 \text{ та } \int_0^1 Rad(n,\theta)Rad(n,\theta)d\theta = 1.$$

- система функцій Радемахера утворює в просторі інтегрованих із квадратом функцій $L_2[0,1)$ неповну систему ортонормованих функцій, оскільки для довільного n не виконується означення повноти системи:

$$\int_0^1 Rad(n,\theta)Rad(1,\theta)Rad(2,\theta)d\theta = 0,$$

тобто існує функція $Rad(1,\theta)Rad(2,\theta)$, яка тотожно не дорівнює нулю на інтервалі $[0,1)$ та ортогональна до всіх функцій системи.

Неповнота системи Радемахера обмежує її застосування для подання інформаційних потоків на основі ортогональних перетворень. Одночасно із широким застосуванням, творенням за допомогою системи Радемахера двійковим кодам властивий недолік, що полягає в неоднозначності формування відліків суміжних кодів при міжрозрядному позиціонуванні. Уникнути такої вади дозволяє перехід до кодів Грея

1.2.2.5. Базис ортогональних функцій Грея.

Екстракція *cos*-складових згідно кожного з порядків n набору дискретно-фазових функцій утворює систему функцій Грея. Система функцій Грея є підмножиною системи дискретно-фазових функцій:

$$Gry(0,\theta) = Dyf(0,\theta + 2^{-1},0);$$
$$Gry(m,\theta) = Dyf(m,\theta,2^{n-m-1}),\ m = 1,2,\ldots,n. \qquad (1.14)$$

Графіки функцій Грея наведено на рис.1.6.

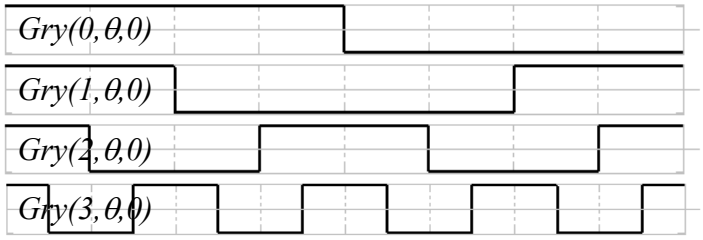


Рис.1.6. Функції Грея.

Система функцій Грея є ортонормованою та неповною, що доводиться наступними викладками.

Відомо, що система Грея є складовою підсистемою функцій Уолша. Оскільки система функцій Уолша є ортонормованою, то і функції Грея, як її підсистема, є ортонормованими, тобто:

$$\int_0^1 Gry(n,\theta)Gry(k,\theta)d\theta = 0,\ \int_0^1 Gry(k,\theta)Gry(k,\theta)d\theta = 1.$$

Функції Грея утворюють у $L_2[0,1]$ неповну систему, оскільки існують функції, які тотожно не дорівнюють нулю та ортогональні до всіх функцій системи, зокрема, для довільного n:

$$\int_0^1 Gry(n,\theta)Rad(2,\theta)d\theta = 0.$$

Неповнота системи Грея обмежує її застосування для розкладання інформаційних потоків та реалізації ТЧП.

Розширення функціональних можливостей при реалізації системних функцій перетворення форми та обробки інформації забезпечує перехід до базису Уолша. Аналітичні залежності процедури переходу базуються на наступній властивості скінченних добутків функцій системи Грея.

Якщо $(k_1,\ldots, k_m)$ і $(l_1,\ldots, l_r)$ дві різні скінченні послідовності, то:

$$\int_0^1 [Gry(k_1,\theta)...Gry(k_m,\theta)][Gry(l_1,\theta)...Gry(l_r,\theta)]d\theta = 0. \qquad (1.15)$$

Для доведення властивості необхідно перепозначити множники в підінтегральному виразі $Gry(j_1, \theta), Gry(j_2, \theta),\ldots, Gry(j_p, \theta)$ $(j_1<j_2<\ldots<j_p)$. Добуток пар функцій $Gry(j_k, \theta)Gry(j_k, \theta)=1$. Добуток інших множників $Gry(j_1, \theta)\, Gry(j_2, \theta)\ldots\, Gry(j_{p-1}, \theta)$ є частково-сталою функцією, кожний з інтервалів сталості якої можна поділити на парне число рівних підінтервалів, на яких $Gry(j_p,\theta)$ набуває почергово значення $+1$ і -1 або -1 і $+1$. Із врахуванням чого:

$$\int_0^1 [Gry(j_1,\theta)...Gry(j_p,\theta)]d\theta = const\int_I Gry(j_p,\theta)d\theta = 0,$$

а тому буде рівним нулю значення інтегралу на інтервалі $[0;1)$. Тобто два різні добутки функцій системи є ортогональними, що і треба довести.

Система функцій Грея є основою кодів Грея. Відповідність між значеннями функцій у точках $\theta_s = s/2^n$, $s=0,1,\ldots,2^n-1$ та їх поданням у коді Грея $\theta_s = h_n h_{n-1}\ldots h_0$ встановлюється співвідношенням:

$$h_k = (1 - Gry(n-k-1,\theta_s))/2, \qquad (1.16)$$

де $k = 0,1,\ldots,n-1$.

Наприклад, чотирьом функціям відповідають елементи кодової матриці розміру 4×8

$Gry(0,\theta)$	$\rightarrow$	0	0	0	0	1	1	1	1
$Gry(1,\theta)$	$\rightarrow$	0	0	1	1	1	1	0	0
$Gry(2,\theta)$	$\rightarrow$	0	1	1	0	0	1	1	0
$Gry(3,\theta)$	$\rightarrow$	0	1	0	1	0	1	0	1
s	$\rightarrow$	0	1	2	3	4	5	6	7.

Кодування Грея дозволяє зменшити похибки формування відліків суміжних кодів унаслідок зміни тільки одного розряду порівняно із застосуванням двійкових кодів. Однак, неповнота систем функцій Радемахера та Грея звужує галузі їх ефективного застосування. Розширення функціональних можливостей при реалізації системних функцій перетворення форми та обробки інформації забезпечує базис Уолша.

1.2.2.6. Базис Уолша.

Властивості систем Радемахера, Грея та відповідних кодів визначають процедуру переходу в базис Уолша *Wal(i,θ),* $i = 0,1,\ldots,2^n - 1$, впорядкований за Уолшем, із системи Радемахера *Rad(n,θ)*. Функції Уолша *Wal(i,θ)* (рис.1.7) визначаються, як добуток функцій Радемахера:

$$Wal(i,\theta) = Rad(1,\theta)^{b_0}\, Rad(2,\theta)^{b_1} \ldots Rad(n,\theta)^{b_{n-1}} = \prod_{k=0}^{n-1} (Rad(k+1,\theta))^{b_k}\ , \quad (1.17)$$

де $i = b_{n-1}b_{n-2}\ldots b_1 b_0$ – подання в коді Грея порядкового номера функції $Wal(i,\theta)$.

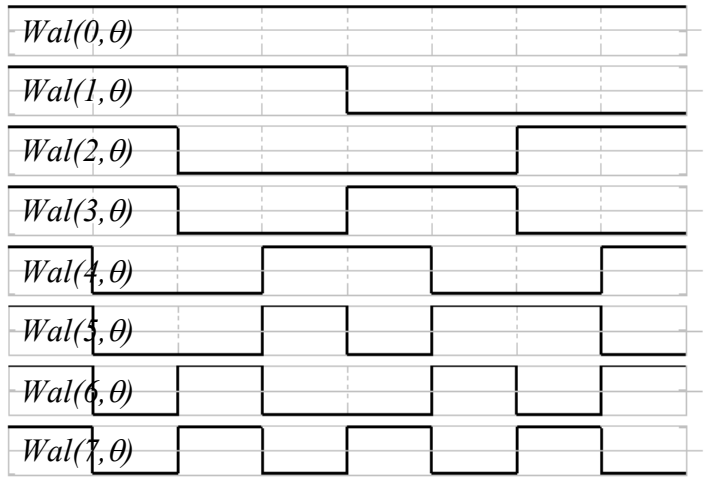


Рис.1.7. Система функцій Уолша.

Система функцій Уолша є ортонормованою, повною та мультиплікативною.

Функції Уолша ортонормовані на інтервалі $0 \le \theta \le 1$:

$$\int_0^1 Wal(k,\theta)Wal(i,\theta)d\theta = \begin{cases} 1 \ \textit{при} \ k = i, \\ 0 \ \textit{при} \ k \ne i. \end{cases}$$

Функції Уолша утворюють мультиплікативну систему:

$$Wal(k,\theta)Wal(i,\theta) = Wal(k \oplus i,\theta).$$

Системи Радемахера та Грея є підсистемами функцій Уолша:

$$Wal\,(2^n\text{–}1,\theta) = Rad(n,\theta),$$
$$Wal\,(2^n,\theta) = Gry(n,\theta).$$

Відомі схеми генераторів функцій Уолша базуються на методі формування функцій Уолша із системи Радемахера, але використання в ньому двійкового коду зумовлює виникнення помилки неоднозначності. На основі доведеної властивості (1.11) ортогональності добутків функцій Грея розроблено альтернативний метод формування функцій Уолша із системи функцій Грея:

$$Wal(i,\theta) = Gry(0,\theta)^{a_0}\, Gry(1,\theta)^{a_1} \ldots Gry(n\text{-}1,\theta)^{a_{n\text{-}1}} = \prod_{k=0}^{n-1} (Gry(k,\theta))^{a_k}, \quad (1.18)$$

де $i = a_{n\text{-}1}a_{n\text{-}2}\ldots a_1 a_0$ – подання в двійковому коді порядкового номера функції Уолша $Wal(i,\theta)$.

У відомих генераторах і перетворювачах функції Уолша формуються згідно (1.17) на основі двійкового коду наступним способом. Функції Радемахера відповідають значенням розрядів двійкового коду:

$$Rad(m,\theta_s) = \overline{r_{n\text{-}m}} \text{-} r_{n\text{-}m} = \begin{cases} 1, \ \textit{якщо} \ r_{n-m} = 0, \\ -1, \ \textit{якщо} \ r_{n-m} = 1, \end{cases} \quad (1.19)$$

де r_m – значення m-го розряду двійкового коду аргумента θ_s, $m=0,1,\ldots,n$.

При підстановці (1.20) у вираз (1.18) функції Уолша визначаються на основі двійкового коду аргументу θ_s:

$$\begin{aligned} Wal(i,\theta_s) &= (\overline{r_{n-1}} - r_{n-1})^{b_0} (\overline{r_{n-2}} - r_{n-2})^{b_{n1}} \ldots (\overline{r_0} - r_0)^{b_{n-1}} = \\ &= (\overline{b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0}) - (b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0) = , \\ &= \overline{\varphi_i}(\theta_s) - \varphi_i(\theta_s) \end{aligned} \quad (1.20)$$

де $\varphi_i(\theta_s) = b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0$.

Із використанням наведеної методики можна визначити спосіб формування функцій Уолша на основі аргументу, поданого в коді Грея.

Функції Грея $Gry(k,\theta_s)$ відповідають значенням n розрядів коду Грея θ_s згідно залежності:

$$Gry(k,\theta_s) = \overline{h_{n-k-1}} - h_{n-k-1} = \begin{cases} 1, & \text{ÿêùî} \quad h_{n-k-1} = 0, \\ -1, & \text{ÿêùî} \quad h_{n-k-1} = 1, \end{cases} \tag{1.21}$$

де h_k – значення *k*-го розряду коду Грея аргументу θ_s; $k = 0,1,\ldots,n-1$.

Функції Уолша визначаються при підстановці функцій Грея з (1.21) у (1.18):

$$Wal(i,\theta_s) = (\overline{h_{n-1}} - h_{n-1})^{a_0} (\overline{h_{n-2}} - h_{n-2})^{a_1} \ldots (\overline{h_0} - h_0)^{a_{n-1}},$$

де $\theta_s = h_{n-1}\ldots h_1 h_0$ – код Грея, $i = a_{n-1}a_{n-2}\ldots a_0$ – подання у двійковому коді числа *i*.

Перетворення добутку в правій частині рівності з використанням основних тотожних співвідношень булевої алгебри дозволяє визначити функції Уолша:

$$Wal(i,\theta_s) = (\overline{a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0}) - (a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0), \tag{1.22}$$

$$Wal(i,\theta_s) = \overline{\varphi_i}(\theta_s) - \varphi_i(\theta_s), \tag{1.23}$$

де $\varphi_i(\theta_s) = a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0$.

Перевагою перетворювача на основі (1.22) є використання кодів Грея, які дозволяють зменшити помилки в засобах перетворення та обробки.

Таким чином, повна система Уолша, яка утворюється із систем Радемахера та Грея, є базисом для виконання ортогональних перетворень і формування функцій Галуа.

1.2.2.7. Базис Крестенсона.

Узагальненням системи функцій Уолша на *p*-значний випадок, де *p* – просте число, є система функцій Крестенсона. Функції Крестенсона (рис.1.8) $\chi_\omega^{(p)}(z)$ є частково-постійними функціями з інтервалом значень аргументів $[0, p^m)$ і визначаються для будь-якого з натуральних ω, m при $0 \le \omega \le p^m - 1$ наступним чином:

$$\chi_\omega^{(p)}(p) = \exp(j\frac{2\pi}{p}\sum_{i=0}^{m-1}\omega^{(m-1-i)}z^{(i)}), \tag{1.24}$$

де $j = \sqrt{-1}$; $\omega^{(i)}, z^{(i)} \in [0,1,...,p-1]$ [3].

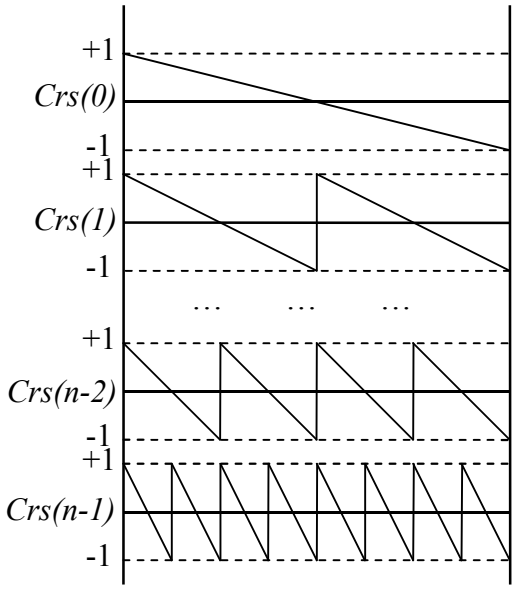


Рис.1.8. Система функцій Крестенсона.

$$\left.\begin{aligned}\omega &= \sum_{i=0}^{m-1} \omega^{(i)} p^{m-1-i,} \\ z &= \sum_{i=0}^{\infty} z^{(i)} p^{m-1-i}\end{aligned}\right\}. \qquad (1.25)$$

Отже, отримується:

$$\chi_{\omega}^{(2)}(z) = W_{\omega}(z)\ , \qquad (1.26)$$

при *p=2* система функцій Крестенсона співпадає з системою функцій Уолша.

Функції Крестенсона при p=3, m=2 приведені у таблиці 1.1. Тут і далі, таблиці 1-7,

$e_1 = -0{,}5(1 - j\sqrt{3}),\ \bar{e}_2 = l_1 = -0{,}5(1 + j\sqrt{3}),$ *äå* $j = \sqrt{-1}$.

Розглянемо підмножину $\left\{K_{r,i}^{(p)}(z)\right\}$ функцій Крестенсона $\left\{K_{r,i}^{(p)}(z)\right\}$, для яких індекс ω містить усього одну p компоненту. Ці функції є узагальненнями функцій Радемахера :

$$K_{r,i}^{(p)}(z) = \chi_{r,p^i}^{(p)}(z) = \exp(j\frac{2\pi}{p} r z^{(i)}) \qquad (1.27)$$

(r=1,2, ... , p – 1, i=0,1, ..., m-1).

Отже,

$$K_{j,i}^{(2)}(z) = R_{i+1}(z) \qquad (1.28)$$

у зв'язку з цим функції $K_{j,i}^{(2)}(z)$ називаються узагальненими функціями Радемахера. Узагальнені функції Радемахера при p=3, m=2 приведені в табл. 1.1

Узагальнені функції Уолша і Радемахера приймають значення з множини $\left\{\exp(j\frac{2\pi}{p}q)\right\}$, $(q=0,1,...,p-1)$.

Можна встановити взаємно-однозначну відповідність *h* між безліччю значень функцій Крестенсона і безліччю чисел $[0,1,...,p-1]$ наступним чином:

$$h\left(\exp(j\frac{2\pi}{p}q)\right)=q,\ (q=0,1,...,p-1)\,. \qquad (1.29)$$

Таблиця 1.1.

$\chi_{\omega}^{(3)}(z)$	z								
	0	1	2	3	4	S	6	7	8
$\chi_0^{(3)}(z)$	1	1	1	1	1	1	1	1	1
$\chi_1^{(3)}(z)$	1	1	1	e_1	e_1	e_1	e_2	e_2	e_2
$\chi_2^{(3)}(z)$	1	1	1	e_2	e_2	e_2	e_1	e_1	e_1
$\chi_3^{(3)}(z)$	1	e_1	e_2	1	e_1	e_2	1	e_1	e_2
$\chi_4^{(3)}(z)$	1	e_1	e_2	e_1	e_2	1	e_2	1	e_1
$\chi_5^{(3)}(z)$	1	e_1	e_2	e_2	1	e_1	e_1	e_2	1
$\chi_6^{(3)}(z)$	1	e_2	e_1	1	e_2	e_1	1	e_2	e_1
$\chi_7^{(3)}(z)$	1	e_2	e_1	e_1	1	e_2	e_2	e_1	1
$\chi_8^{(3)}(z)$	1	e_2	e_1	e_2	e_1	1	e_1	1	e_2

Тоді, як видно з (1.29), після перетворення *h* функції $K_{1,i}^{(p)}(z)$ співпадають з відповідним розрядом $z^{(t)}$ *p*-того коду аргументу *p*.

Функції Крестенсона можуть бути виражені через узагальнені функції Радемахера.

З (1.22) і (1.25) отримується:

$$\chi_{\omega}^{(p)}(z)=\prod_{i=0}^{m=1}(K_{1,i}^{(p)}(z))\omega^{(m-1-i)} \qquad (1.30)$$

Формула (1.30) є узагальненням і показує зв'язок функцій Радемахера і Уолша.

$K_{r,i}^{(3)}(z)$	z								
	0	1	2	3	4	S	6	7	8
$K_{0,0}^{(3)}(z)$	1	1	1	1	1	1	1	1	1
$K_{1,0}^{(3)}(z)$	1	1	1	e_1	e_1	e_1	e_2	e_2	e_2
$K_{2,0}^{(3)}(z)$	1	1	1	e_2	e_2	e_2	e_1	e_1	e_1
$K_{1,1}^{(3)}(z)$	1	e_1	e_2	1	e_1	e_2	1	e_1	e_2
$K_{2,1}^{(3)}(z)$	1	e_1	e_1	1	e_2	e_1	1	e_2	e_1

До властивостей функцій Крестенсона можна віднести повноту і ортогональність.

Теорема. Множина p^m функцій Крестенсона утворює повну ортогональну систему функцій в просторі частково-постійних функцій, визначених на $[0,p^m)$, причому:

$$\sum_{z=0}^{p^m-1} x_t^{(p)}(z)\overline{x_q^{(p)}(z)} = \begin{cases} p^m \text{ при } t = q \\ 0 \text{ при } t \neq q \end{cases} \qquad (1.31)$$

і, якщо $\Phi(z)$ – частково-постійна функція і

$$\sum_{z=0}^{p^m-1} \Phi(z)\overline{x_q^{(p)}(z)} = 0 \quad (\omega = 0,1,..., p^m - 1),$$

то $\Phi(z) \equiv 0$

З (1.31) при $t=0$ випливає:

$$\sum_{z=0}^{p^m-1} x_\omega^{(p)}(z) = 0 \text{ при } \omega \neq 0 \qquad (1.32)$$

Теорема. Нехай функція $\Phi(z)$ – частково-постійна, яка представляє деяку систему p - тих функцій від m аргументів. Тоді:

$$\Phi(z) = \sum_{\omega=0}^{p^m-1} S(\omega)x_\omega^{(p)}(z), \qquad (1.33)$$

Таблиця 1.2.

z,ω	$z^{(0)}$	$z^{(1)}$	$f^{(0)}(z)$	$f^{(1)}(z)$	$\Phi(z)$	$9S(\omega)$
0	0	0	0	2	2	36
1	0	1	2	2	8	0
2	0	2	*0*	2	2	0
3	1	0	1	0	3	0
4	1	1	0	1	1	$1,5-1,5\sqrt{3}\,j$
5	1	2	2	2	8	$-10,5-7,5\sqrt{3}\,j$
6	2	0	2	1	7	0
7	2	1	1	0	3	$-10,5+7,5\sqrt{3}\,j$
8	2	2	0	2	2	$1,5+1,5\sqrt{3}\,j$

де

$$S(\omega) = p^{-m}\sum_{z=0}^{p^m-1}\hat{O}(z)\overline{x_\omega^{(p)}(z)}. \qquad (1.34)$$

Отже, якщо початкова система набуває ненульових значень не більше ніж в p^m точках, то і спектр по Крестенсону $S(\omega)$ набуває ненульових значень також не більше ніж в p^m точках. Це дає можливість відновлення початкової системи p-тих функцій по її спектру по Крестенсону.

Приклад. Нехай система двох трійкових функцій від двох аргументів задається табл.1.2 ($p=3,m=2$). Частково постійна функція $\Phi(z)$, її спектр по Крестенсону S(ω) приведені в таблицю. 1-8. Таким чином для цього прикладу:

$\Phi(z)=3^{-2}\ (36+1,5(1-j\sqrt{3}\,))x^{(3)}{}_4\ (z)+(-10,5-j7,5\sqrt{3}\,)x^{(3)}{}_5\ (z)\ +(-10,5-j7,5\sqrt{3}\,)x^{(3)}{}_7\ (z)+(1,5+j1,5\sqrt{3}\,)x^{(3)}{}_8\ (z)$

Теорема (*Симетрія індексу і аргументу)*. Для будь-кого ω,z $\in[0,1,\ldots,\mathrm{p^m}\text{-}1]$, $x_\omega^p(z) = x_z^p(\omega)$.

Теорема (*Зрушення аргументу)*. Для будь-кого

$$\omega,z,\tau \in [0,1,\ldots,p^m\text{-}1], \qquad (1.35)$$

$$x_\omega^{(p)}(z\oplus\tau) = x_\omega^{(p)}(z)x_\omega^{(p)}(\tau)\ (\mathrm{mod}\ p), \qquad (1.36)$$

з попередніх теорем слідує, що безліч функцій Крестенсона замкнута відносно операції множення.

Ізоморфізм між лінійними функціями p- тої логіки і функціями Крестенсона. Функція $f(z^{(0)}, z^{(1)}, \ldots, z^{(m-1)})$ p - значної логіки називається лінійною, якщо її можна представити згідно співвідношення:

$$f(z^{(0)},z^{(1)},\ldots,z^{(m-1)}) = \bigoplus_{i=0}^{m-1} c_i z^{(i)}\ (\mathrm{mod}\ p) \qquad (1.37)$$

де $c_i \in [0,1,...,p-1]$.

Лінійні функції утворюють комутативну групу з груповою операцією «сума по модулю *p*» і з числом елементів p^m. Функції Крестенсона $x_\omega^{(p)}(z)$ створюють групу відносно операції множення.

Теорема. Мультиплікативна група функцій Крестенсона ізоморфна групі лінійних функцій *p*-значної логіки. Ізоморфізм *h* задається співвідношенням:

$$h(x_\omega^{(p)}(z)) = \bigoplus_{i=0}^{m-1} \omega^{(m-1-i)} z^{(i)} \pmod p \tag{1.38}$$

Цей ізоморфізм *h* є продовженням ізоморфізма між узагальненими функціями Радемахера і компонентами *p* - того розкладання аргументу і надалі використовуватиметься при аналізі і синтезі схем, що реалізовують системи функцій *p*-значної логіки.

Розглянемо тепер методи обчислення спектру по Крестенсону. Один метод, природно, визначається при $\psi_\omega(z) = x_\omega^{(p)}(z)$. Інший метод полягає в узагальненні методу обчислення спектру по Уолшу через матриці Адамара. Будується матриця $x_\omega^{(p)}$, ω - рядок якої буде

$x_\omega^p(0),\ x_z^p(1),...,x_z^p(p^m-1)(\omega = 0,1,...,p^m-1)$.

При *m = 1* і *m = 2* для *p=3* $x_1^{(3)}$ *і* $x_1^{(3)}$ має вигляд:

$$x_1^{(3)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & e_1 & e_2 \\ 1 & e_2 & e_1 \end{pmatrix} \tag{1.39}$$

$$x_2^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & e_1 & e_1 & e_1 & e_2 & e_2 & e_2 \\ 1 & 1 & 1 & e_2 & e_2 & e_2 & e_1 & e_1 & e_1 \\ 1 & e_1 & e_2 & 1 & e_1 & e_2 & 1 & e_1 & e_2 \\ 1 & e_1 & e_2 & e_1 & e_2 & 1 & e_2 & 1 & e_1 \\ 1 & e_1 & e_2 & e_2 & 1 & e_1 & e_1 & e_2 & 1 \\ 1 & e_2 & e_1 & 1 & e_2 & e_1 & 1 & e_2 & e_1 \\ 1 & e_2 & e_1 & e_1 & 1 & e_2 & e_2 & e_1 & 1 \\ 1 & e_2 & e_1 & e_2 & e_1 & 1 & e_1 & 1 & e_2 \end{bmatrix} \tag{1.40}$$

Усі властивості функцій Крестенсона можуть бути сформульовані з використанням матриць $x_m^{(p)}$, подібно до того як це мало місце для функцпй Уолша і матриць Адамара.

Представимо початкову функцію і її спектр по Крестенсону у вигляді векторів $\vec{\Phi} = [\Phi(0), \Phi(1), ..., \Phi(p^m - 1))$, $\vec{S} = (S(0), S(1), ..., S(p^m - 1))]$

Тоді з (1.34) випливає, що:

$$\vec{S} = p^{\overline{-m x_m^{(p)}}} \vec{\Phi} . \qquad (1.41)$$

Формула (1-50) дає простий і зручний при невеликих p і m спосіб побудови спектру. Обчислення спектру по Крестенсону по формулі (1-50) вимагає p^{2m} операцій додавання і віднімання (по p^m операцій на коефіцієнт). Наведем тепер спосіб обчислення спектру по Крестенсону, який вимагає мінімального числа операцій, рівного $mp^m(p-1)$. Цей спосіб буде узагальненням на p - значний випадок.

Якщо A симетрична матриця розміру $p^m \times p^m$ то будемо позначати $\overleftarrow{A}$ матрицю, ω -й рядок і стовпець яки є ω рядок і стовпець матриці A, де, якщо

$$\omega = \sum_{i=0}^{m-1} \omega^{(m-1-i)} p^i , \overleftarrow{\omega} = \sum_{i=0}^{m-1} \omega^{(i)} p^{(i)} .$$

Утворюється матриця:

$$A_m^{(p)} \begin{bmatrix} e_0^0 e_{p-1}^0 ... e_{p-1}^0 & 0 & \\ 0 & e_0^0 e_1^0 ... e_{p-1}^0 & 0 \\ e_0^1 e_1^1 ... e_{p-1}^0 & 0 & e_0^0 e_1^0 ... e_{p-1}^0 \\ 0 & e_0^1 e_1^1 ... e_{p-1}^1 & 0 \\ & 0 & e_0^1 e_1^1 ... e_{p-1}^1 \\ e_0^{p-1} e_{p-1}^{p-1} ... e_{p-1}^{p-1} & 0 & \\ 0 & e_0^{p-1} e_1^{p-1} ... e_{p-1}^{p-1} & 0 \\ & 0 & e_0^{p-1} e_1^{p-1} ... e_{p-1}^{p-1} \end{bmatrix} \} p^{m-1}$$

$$e_q = \exp(j \frac{2\pi}{p} q) \quad (q = 0,1,..., p-1) .$$

Теорема.

$$\overline{x_m^{(p)}} = \overleftarrow{(\overline{A_m^{(p)}})^m} \qquad (1.42)$$

Теорема узагальнює наслідок з теореми 1-6, визначає факторизацію матриці $\overline{x_m^{(p)}}$ і разом з формулою **(**1.42**)** визначає «швидке перетворення

Адамара – Крестенсона», при використанні якого для обчислення спектру по Крестенсону вимагається $mp^m(p\text{-}1)$ операцій додавання і віднімання (множення $\vec{\Phi}$ на матрицю $\overline{A_m^{(p)}}$ вимагає p^m операцій).

Кожен з коефіцієнтів розкладання системи Функцій p-значної логіки по базису Крестенсона враховує поведінку системи на всьому інтервалі завдання.

1.2.2.8. Базис та кодові системи Галуа.

Перехід до різних упорядкувань функцій у системі Галуа здійснюється з базису Уолша з упорядкуванням функцій за рекурсивним законом. За n-розрядними фрагментами рекурсивної послідовності, яка утворюється відповідно до породжуючого вектора поля Галуа $GF(2^n)$, згідно відображення через систему функцій Радемахера формуються номери функцій Уолша та Галуа в системі.

Наприклад, у полі $GF(2^3)$ існують породжуючі вектори 1011 та 1101. Для даного поля $GF(2^3)$ рекурсивні послідовності $v_0, v_1, v_2, v_3, v_4, \ldots$ формуються з початкового вектора $(v_0 v_1 v_2) = (111)$ за правилами:

1) $1101 \rightarrow v_{i+3} = v_i \oplus v_{i+1}$:

$v_0, v_1, v_2, v_0 \oplus v_1, v_1 \oplus v_2, v_0 \oplus v_1 \oplus v_2, v_0 \oplus v_2, v_0, v_1, v_2, \ldots;$

2) $1011 \rightarrow v_{i+3} = v_i \oplus v_{i+2}$:

$v_0, v_1, v_2, v_0 \oplus v_2, v_0 \oplus v_1 \oplus v_2, v_0 \oplus v_1, v_1 \oplus v_2, v_0, v_1, v_2, \ldots.$

Для початкового вектора $(v_0 v_1 v_2) = (111)$ утворена на основі породжуючого вектора 1101 рекурсивна послідовність {0 0 0 1 0 1 1 1}, визначає наступне рекурсивне впорядкування номерів функцій Уолша в системі {0 1 2 5 3 7 6 4}.

0	0	0	1	0	1	1	1	0	0		
0	0	0								$\rightarrow$	0
	0	0	1							$\rightarrow$	1
		0	1	0						$\rightarrow$	2
			1	0	1					$\rightarrow$	5
				0	1	1				$\rightarrow$	3
					1	1	1			$\rightarrow$	7
						1	1	0		$\rightarrow$	6
							1	0	0	$\rightarrow$	4

Утворена на основі породжуючого вектора 1011 рекурсивна послідовність {0 0 0 1 1 1 0 1} визначає інше рекурсивне впорядкування номерів функцій Уолша: {0 1 3 7 6 5 2 4}.

Із рекурсивно впорядкованої системи Уолша відповідно впорядковані перші *n* функцій Галуа формуються згідно співвідношення:

$$Gal(n,\theta,i) = Wal(Ent(2^n\theta), \frac{2^{i+1}-1}{2^n}), \qquad (1.24)$$

де $i = 0,1,\ldots,2^n-1$, *Ent* – функція виділення цілої частини.

Проведені дослідження встановили можливість формування функцій Галуа із систем Радемахера та Грея. Згідно співвідношень (1.17) та (1.18) перші *n* функцій Галуа в системі подаються у вигляді добутку функцій Радемахера та Грея:

$$Gal(n,\theta,i) = \prod_{k=0}^{n-1}(Rad(k+1, \frac{2^{i+1}-1}{2^n}))^{h_k} = \prod_{k=0}^{n-1}(Gry(k+1, \frac{2^{i+1}-1}{2^n}))^{r_k}, \quad (1.25)$$

де $h_{n-1}h_{n-2}\ldots h_0$ – запис у коді Грея числа *q*, двійковий код якого є *n*-розрядним фрагментом $v_i v_{i+1} v_{i+2}\ldots v_{i+n-1}$ рекурсивної послідовності $v_0, v_1, v_2, \ldots$; $r_{n-1}r_{n-2}\ldots r_0$ – двійковий код, який є *n*-розрядним фрагментом $v_i v_{i+1} v_{i+2}\ldots v_{i+n-1}$ рекурсивної послідовності $v_0, v_1, v_2, \ldots$.

Повний набір 2^n функцій рекурсивної системи Галуа $Gal(n,\theta,i)$ отримують із перших *n* функцій системи процедурою рекурсивного зсуву на $\Delta\theta = \frac{1}{2^n}$ згідно другої діагоналі кожної наступної функції відносно попередньої:

$$Gal(n,\theta,i+1) = Gal(n,\theta+\Delta\theta,i). \qquad (1.26)$$

Впорядкування функцій Галуа в наборі відповідає синтезованому за породжуючим вектором упорядкуванню функцій Уолша.

Процедура переходу від дискретних значень функцій Уолша до дискретних значень функцій Галуа подається матричною операцією:

$$\|Gal\| = \|W\| \cdot \|R\|,$$

де $\|Gal\|$ – матриця розміру $N \times n$ системи Галуа; $\|W\|$ – матриця розміру $N \times N$ рекурсивно впорядкованих функцій Уолша; $\|R\|$ – матриця розміру $N \times n$ відображеної вагової мережі Радемахера.

Для прикладу, матрична операція переходу від функцій Уолша до функцій Галуа та матриця розміру *8×8* дискретних значень функцій Галуа в полі $GF(\binom{3}{2})$ з породжуючим вектором 1101 згідно процедури рекурсивного розширення подаються відповідно:

$$\begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{Vmatrix} \cdot \begin{Vmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{Vmatrix} = \begin{Vmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & 1 \end{Vmatrix} \longrightarrow \begin{Vmatrix} 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \end{Vmatrix}$$

Графіки функцій Галуа $Gal(n,\theta,i)$ з породжуючим вектором 1101 при *n=3* наведено на рис.1.9.

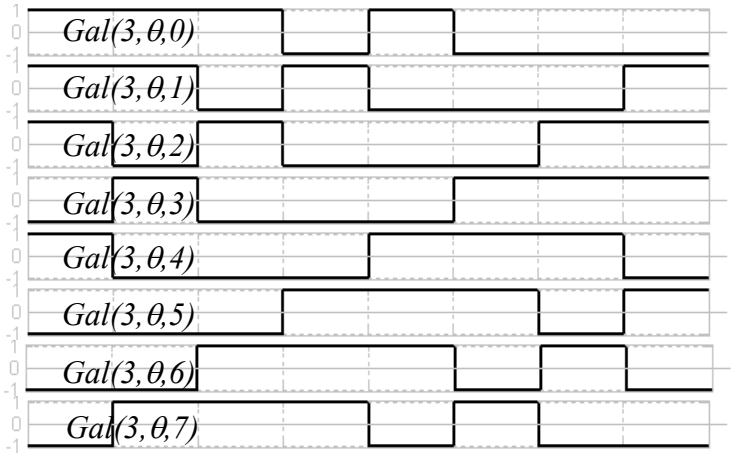


Рис.1.9. Функції Галуа $Gal(3,\theta,i)$ з породжуючим вектором 1101.

При виконанні перетворень використовуються наступні властивості системи функцій Галуа.

1. У системі 2^n функцій n-го порядку кожна підсистема із *n* функцій $\{Gal(n,\theta,i),\ Gal(n,\theta,i+1),\ Gal(n,\theta,i+2),\ldots,\ Gal(n,\theta,i+n-1)\}$ ортогональна.

2. Симетрія індексу та аргументу. Елементи матриці системи Галуа симетричні відносно головної діагоналі:

$$Gal(n,\theta_s,i) = Gal(n,\frac{i}{2^n},s),$$

де $i,s \in \{0,1,\ldots.2^n-1\}$.

Таким чином, матриці системи Галуа є ганкелевими антициклічними, оскільки при $i+j=k+l$ $G_{ij}=G_{kl}$. Якщо рекурсивний зсув у (1.26) здійснюється

згідно головної діагоналі та у формулі (1.26) $\Delta\theta = -\frac{1}{2^n}$, то матриці є тоепліцевими циклічними (циркулянтними), оскільки $i+j=k+l$ $G_{ij}=G_{kl}$.

3. Міри довжин інтервалів, на яких $Gal(n,\theta_s,i)=1$ і $Gal(n,\theta_s,i)=-1$ однакові, отже:

$$\int_0^1 Gal(n,\theta,i)d\theta = \sum_{s=0}^{2^n-1} Gal(n,\theta_s,i) = 0, \qquad (1.27)$$

тобто множина функцій Галуа задовольняє необхідну умову для вейвлет-функцій.

При дискретизації за параметром часу перших n функцій Галуа та здійсненні бінарної заміни значень функцій 1 на 0, –1 на 1, згідно виразу:

$$g_k(\theta_s) = (1 - Gal(n-k-1,\theta_s))/2, \qquad (1.28)$$

одержують матрицю кодових елементів Галуа розміру $n \times N$, $k = 0,1,\ldots,n-1$.

Наприклад, при N=8 рядки матриці кодових елементів Галуа з породжуючим вектором 1101 відповідатимуть таким функціям:

$Gal(3,\theta,0)$	$\rightarrow$	0	0	0	1	0	1	1	1
$Gal(3,\theta,1)$	$\rightarrow$	0	0	1	0	1	1	1	0
$Gal(3,\theta,2)$	$\rightarrow$	0	1	0	1	1	1	0	0
s	$\rightarrow$	0	1	2	3	4	5	6	7

Елементи матриці кодових елементів Галуа з породжуючим вектором 1011 відповідатимуть наступним функціям:

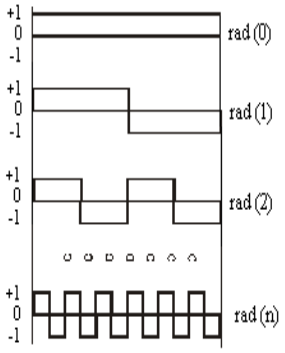
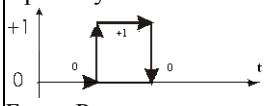
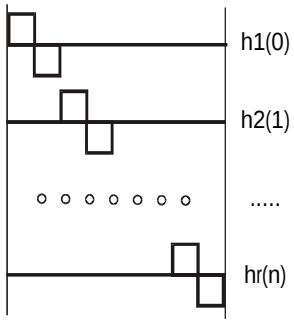
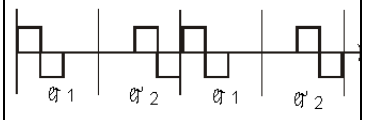
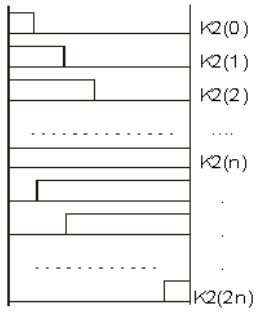
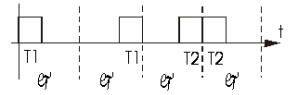
$Gal(3,\theta,0)$	$\rightarrow$	0	0	0	1	1	1	0	1
$Gal(3,\theta,1)$	$\rightarrow$	0	0	1	1	1	0	1	0
$Gal(3,\theta,2)$	$\rightarrow$	0	1	1	1	0	1	0	0
s	$\rightarrow$	0	1	2	3	4	5	6	7

При дискретизації системи N функцій Галуа $\{Gal(n,\theta,i)\}$, $i = 0,1,\ldots 2^n - 1$ та перетворенні значень функцій згідно (5.30) отримують повну матрицю кодових елементів Галуа розміру $N \times N$, впорядкованих із поелементним рекурсивним зсувом згідно другої діагоналі матриці Галуа. Номер s повідомлення однозначно визначається n-координатним вектором $n = \log_2 N$.

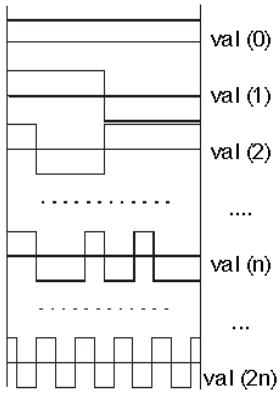
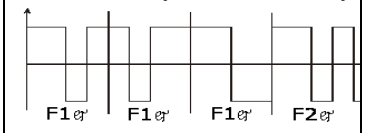
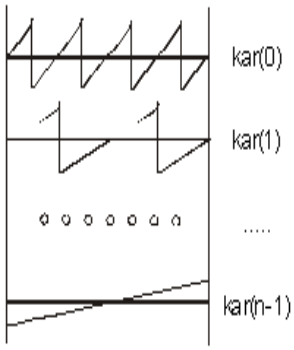
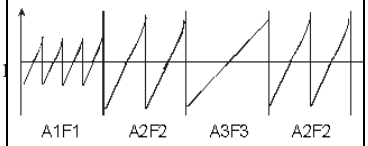
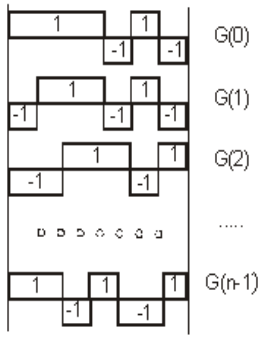
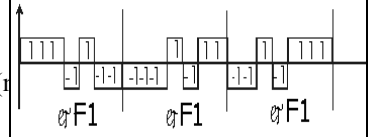
Представлення систем ортогональних функцій різних ТЧБ подані у табл.1.3.

Таблиця 1.3.

Представлення теоретико-числових базисів.

Базис	Представлення базису	Базисна функція та об'єм матриці V	Модуляція сигналу та спектр
1	2	3	4
Радема-хера		Rad(n,θ)=sign[$2^n\pi\cdot$ θ] V=N·$\log_2$ N	Базисні функції Радемахера є основою для модуляції Прямокутних сигналів  Базис Радемахера породжує двійкову систему числення і двійкові коди
Хаара		$Har(n,\theta,i) = sign[\sin(i\cdot 2^n\pi,\theta)]$ V=N^2	Використовується при фазовій модуляції сигналу.  В даному базисі використовуються розрядно-позиційні коди.
Крейга		Crg(n,θ)=sign[sin((2^n-1)·π·θ)] $V=\frac{N^2}{2}$	Породжує тривалісні і фазовіметоди модуляції сигналу. 

продовження таблиці 1.3

1	2	3	4
Уолша		Had (h,x) = $\prod_{i=1}^{\hat{e}}[\mathrm{ri}\,(\mathrm{x})]\mathrm{hi}$	Породжує частотно-фазові методи модуляції сигналу.  Базис Уолша має найбільш широкий спектр сигналу
Крестенсона		$N_i = res\sum_{i=1}^{n}(B_i \cdot b_i)$ $V = \sum_{i=1}^{m}\log_2(P_i)$	Даний базис породжує амплітудно-частотні методи модуляції.  Базис представлений трикутними функціями. Спектр сигналів такого базису є експоненціальний.
Галуа		$N_j=f(C_{j-n-1},\ldots C_{j-1},C_j)$, $C_j = \sum_{j=0}^{n=1} C_{j-1}\cdot A\cdot(r$ V=N	Базис Галуа породжує коди поля Галуа і систему числення Галуа.  Модуляція в базисі як фазова, так і частотна.

З метою оцінки ефективності кодування даних на основі різних ТЧБ доцільно провести аналіз кодових матриць, які породжують різні системи числення.

При цьому важливою характеристикою кожного базису є об’єм його кодової матриці M_j та число активних елементів m_j (рис.1.10), що визначає

характеристики надлишковості представлення інформації на основі аналітичної оцінки:

$$V_i = n_i \cdot N_i,$$

де n_j – розрядність числа; N_i – число незалежних кодових значень.

$$M_{Uni} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} \quad а)$$

$$M_{har} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} \quad б)$$

$$M_{Gr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{vmatrix} \quad в)$$

$$M_{Rad} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} \quad г)$$

$$M_{LibCr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & & 0 & 1 \end{vmatrix} \quad д)$$

$$M_{Cres} = \begin{vmatrix} P_1 & P_2 & \dots & P_n \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ 0 & 3 & \dots & 3 \\ 1 & 4 & \dots & 4 \\ 2 & 0 & \dots & 5 \\ 0 & 1 & \dots & 6 \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_n \end{vmatrix} \quad е)$$

$$M_{Gal} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix} \quad є)$$

Рис.1.10. Кодові матриці дискретних базисів:
а) унітарного; б) Хаара; в) Грея; г) Радемахера; д) Крейга; е) Крестенсона; є) Галуа.

Кожен з названих базисів характеризується визначеним об'ємом кодової матриці для представлення даних. При цьому найбільш надлишковим базисом є унітарний, в якого кодова матриця $V = N^2$, а число активних кодових елементів $n = N^2/2$, де N – діапазон кодування даних. Аналогічну надлишковість забезпечує базис Хаара, в два рази меншу надлишковість забезпечує базис Крейга, тобто $V = N^2/4$, а $n = N^2/8$. Максимально широке застосування для кодування даних в сучасних КС отримали базиси Радемахера та Крестенсона, в яких $V = N \log_2 N$. Дані

базиси відповідно породжують двійкову систему числення та систему числення залишкових класів.

Базис Уолша максимально широко використовується в сучасних телекомунікаційних КС. Даний базис породжує систему ортогональних шумоподібних сигналів, які використовуються в сотових системах мобільного зв'язку.

Найменшу надлишковість кодування даних забезпечує базис Галуа, кодова матриця якого $V = N$, а $n = N/2$.

Згідно викладеного, характеристики ТЧБ кодування даних, як системного об'єкта, подані в табл.1.4.

Таблиця 1.4.

Характеристики потоків даних

Формувачі вхідних та вихідних інформаційних сигналів даних	Характеристики інформаційних потоків даних
Унітарний базис	$V = N^2;\ n = N^2/2$
Базис Хаара	$V = N^2,\ n = N$
Базис Крейга	$V = N^2/4,\ n = N^2/8$
Базис Радемахера	$V = N \cdot \log_2 N,\ n = \frac{N}{2}\log_2 N$
Базис Крестенсона	$V = N \cdot \log_2 N$
Базис Уолша	$V = N^2, n = N^2/2$
Базис Галуа	$V = N,\ n = N/2$

Світовий досвід створення процесорів для комп'ютерних систем за останні 50 років, поряд з застосуванням теоретико-числового базису (ТЧБ) Радемахера, який породжує двійкову систему числення, демонструє тенденцію все ширшого застосування інших ТЧБ, в тому числі: унітарного, Хаара, Крестенсона та Галуа. Реалізація спеціалізованих, сигнальних, комутаційних та проблемно-орієнтованих процесорів цифрової обробки даних часто виконується на базі сумісного використання комбінацій названих ТЧБ, наприклад Радемахера-Хаара, Крестонсона-Галуа та ін.

Перспективним напрямком розвитку теорії та технологій побудови спеціалізованих програмно-апаратних комп'ютерних засобів є реалізація супершвидкодіючих мультибазисних RCG-процесорів на основі базисів Радемахера, Крестенсона і Галуа. Відомі успішні спроби розвитку теорії та техніки побудови матричних процесорів на основі двовимірних базисів Радемахера та Галуа, а також конвеєрних спецпроцесорів у базисі Галуа.

Спостережувані тенденції розвитку теорії методології та техніки процесорів комп'ютерних систем обумовлені теоретичним та ідейним

насиченням можливостей застосування базису Радемахера для побудови арифметико-логічних компонентів процесорів, до яких ставляться все жорсткіші вимоги щодо швидкодії, покращення регулярності структури та розширення функціональних можливостей.

У зв'язку з цим існує проблема глибокого дослідження характеристик «нерадемахівських» ТЧБ та граничних можливостей їх застосування для реалізації компонентів як спеціалізованих, так і універсальних процесорів. При цьому перспективним, крім найбільш сьогодні масового одновимірного (векторного) представлення чисел та виконання арифметико-логічних операцій у базисі Радемахера перспективним є застосування двовимірних систем числення, вертикальної інформаційної технології у базисі Галуа та різних форм багатовимірного представлення чисел у вигляді залишків різних форм системи залишкових класів базису Крестенсона.

1.3. Числові послідовності та функції.

Числовими функціями називають такі функції, які набувають цілих значень або визначені для цілих значень аргументу.

1.3.1. Числова функція [x] і її застосування.

Важливу роль у теорії чисел відіграє функція $[x]$; вона визначається для всіх дійсних x і є найбільшим цілим числом, що не перевищує $x : x-1<[x]\le x$.

Ця функція називається цілою частиною від x (або антьє від x). Зокрема $[0]=0$, $[2]=2$, $[3{,}7]=3$, $[-1{,}2]=-2$, $\lfloor\sqrt{3}\rfloor=1$, $[-\pi]=-4$ і т. д. Отже, ця функція набуває тільки цілих значень при довільних дійсних значеннях аргументу x.

Очевидно маємо:

$[x]\le x<[x]+1$,

або

$x=[x]+0$,

де $0\le\theta<1$.

Число θ, визначене останньою формулою, називається дробовою частиною x і позначається символом $\{x\}$, так, що $\{x\}=x-[x]$; зокрема $\{6\}=0$, $\{2{,}8\}=0{,}8$, $\{2\}=0$, $\{-5{,}67\}=0{,}33$ і т. д.

Згідно з означенням $\{x\}$ є завжди невід'ємним числом, меншим від одиниці, тобто $0\le\{x\}<1$.

З означення функції $[x]$ випливають такі її основні властивості:

1. Якщо $x = n + 0$, де n - ціле і $0 \le \theta < 1$, то $n = [x]$.

Ця властивість випливає з нерівностей:

$0 \le x - n < 1$, або $x - 1 < n \le x$.

2. $[a + b] \ge [a] + [b]$.

Згідно означень $a + b = [a] + [b] + \{a\} + \{b\}$. Тут можливі два випадки: по-перше, $0 \le \{a\} + \{b\} < 1$; тоді очевидно, що $[a + b] = [a] + [b]$; по-друге, $1 \le \{a\} + \{b\} < 2$; у цьому разі отримується $[a + b] > [a] + [b]$. Отже, в будь-якому випадку $[a + b] \ge [a] + [b]$.

3. Якщо a - дійсне додатне число і b – натуральне число, то натуральних чисел, які не перевищують a і діляться на b, буде точно $\left[\frac{a}{b}\right]$.

Справді, нехай числами, кратними b, і такими, що не перевищують a, будуть k чисел: $b, 2b, 3b \ldots , kb$. Тоді буде справедлива нерівність: $kb \le a < (k+1)b$, звідки $k \le \frac{a}{b} < k + 1$, тобто ліва нерівність: $kb \le a < (k+1)b$, звідки $k \le \frac{a}{b} < k + 1$, тобто, $k = \left[\frac{a}{b}\right]$.

4. Якщо $a > 0$ – будь-яке ціле число і b – натуральне число, то $\left[\frac{[a]}{b}\right] = \left[\frac{a}{b}\right]$.

Справді, між $[a]$ і a немає натуральних чисел і тому кількість чисел, кратних b, і таких, що не перевищують [a] і відповідно a, буде однаковою. За властивістю 3 в першому випадку їх буде, $\left[\frac{[a]}{b}\right]$, а в другому - $\left[\frac{a}{b}\right]$. Отже: $\left[\frac{[a]}{b}\right] = \left[\frac{a}{b}\right]$.

Щоб показати важливість запровадженої функції, розглянемо приклади її застосувань.

Теорема. Показник, з яким дане n просте число p входить до добутку $n!$, дорівнює:

$\left[\frac{n}{p}\right] + \left[\frac{n}{p^2}\right] + \ldots + \left[\frac{n}{p^k}\right]$, де $p^k \le n$, але вже $p^k + 1 > n$.

Справді, на підставі властивості 3, число співмножників добутку $n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n$, кратних p, дорівнюватиме $\frac{n}{p}$; ці співмножники будуть:

$p, 2p, ..., \left[\frac{n}{p}\right]p$. Слід зазначити, що інші числа цього добутку на p не діляться. Отже, поява числа p в канонічному розкладі $n!$ визначається добутком $M = p \cdot 2p \cdot 3p \cdots \left[\frac{n}{p}\right]p = p^{\left[\frac{n}{p}\right]} \cdot 1 \cdot 2 \cdot 3 \cdots \left[\frac{n}{p}\right]$.

Позначивши через $\left[\frac{n}{p}\right] = n_1$, тоді $M = 1 \cdot 2 \cdot 3 \cdots n_1 \cdot p^{n1}$. Серед множників $1,2...n_1$ можуть бути числа, які діляться на $p: p, 2p.3p...., \left[\frac{n_1}{p}\right]p$. Їх добуток дорівнює $1 \cdot 2 \cdot 3 \cdots \frac{n_1}{p} \cdot p^{\frac{n_1}{p}}$, або якщо $n_2 = \left[\frac{n_1}{p}\right] = \left[\frac{n}{p} : p\right] = \left[\frac{n}{p^2}\right]$, тоді згідно властивості 4 дістанемо:

$M = M_1 \cdot 1 \cdot 2 \cdot 3 \cdots n_2 \cdot p^{n_1+n_2}$,

де M_1 – добуток множників, що не діляться на p. Якщо $n_2 < p$, то процес закінчено; якщо $n_2 \geq p$, продовжується далі.

Міркуючи аналогічно, дістанемо:

$M = M_2 \cdot 1 \cdot 2 \cdot 3 \cdots n_3 \cdot p^{n_1+n_2+n_3}$,

де $n3 = \left[\frac{n_2}{p}\right] = \left[\frac{n}{p^3}\right]$ і т. д.

Очевидно, що цей процес скінчений, бо $n > n_1 > n_2 > \cdots$, і при досить великому k виявиться, що $n_k < p$ і $\left[\frac{n_k}{p}\right] = \left[\frac{n}{p^{k+1}}\right] = 0$.

Отже, $M = M_{k-1} \cdot 1 \cdot 2 \cdot 3 \cdots n_k \cdot p^{n_1+n_2+n_3+...+n_k}$.

Серед множників $1,2,..., n_k$ немає таких, що діляться на p, бо $n_k < p$; M_{k-1} також не містить множників, кратних p, отже, до канонічного розкладу $n!$ просте число p ввійде з показником, який дорівнює:

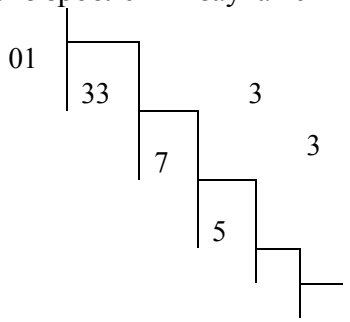
$n_1 + n_2 + \cdots + n_k = \left[\frac{n}{p}\right] + \left[\frac{n}{p^2}\right] + \cdots + \left[\frac{n}{p^k}\right] = \sum_{s=1}^{k} \left[\frac{n}{p^s}\right]$, що й треба було показати.

На практиці обчислення краще проводити за формулою: $n_s = \left[\frac{n_{s-1}}{p}\right]$,

тобто $\left[\frac{n}{p_s}\right] = \left[\left[\frac{n}{p^{s-1}}\right] : p\right]$.

Приклад. Знайти показник степеня, з яким число 7 входить до добутку 701!

Обчислення , згідно із зробленим зауваженням, за такою схемою:

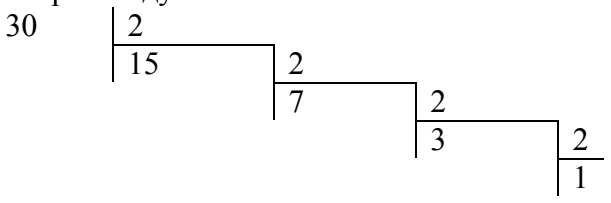


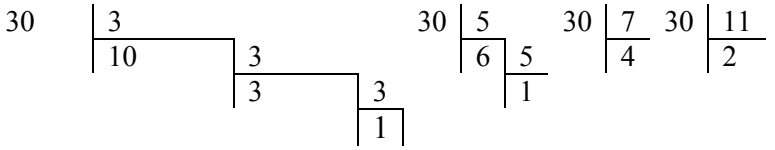
Додаючи частки знайдемо, що шуканий показник дорівнює $233 + 77 + 25 + 8 + 2 = 345$.

Зауваження. Ця теорема, дає можливість знаходити канонічний розклад числа $n!$.

Приклад. Знайти канонічний розклад числа $30! = 1 \cdot 2 \cdot 3 \cdots 30$.

Очевидно, що до канонічного розкладу $30!$ входять тільки прості числа, менші за 30. Знайдемо з якими показниками вони входитимуть до цього розкладу:





Маємо $30! = 2^{26} \cdot 3^{14} \cdot 5^7 \cdot 7^4 \cdot 11^2 \cdot 13^2 \cdot 17 \cdot 19 \cdot 23 \cdot 29$.

Теорема. Якщо $a, b, ..., l, n$ - натуральні числа і $n \geq a + b + ... + l$, то $\frac{n!}{a!b!..l!}$ – натуральне число.

Приклад. Якщо $m > n$, то $\frac{n(n-1)...(n-m+1)}{1 \cdot 2 \cdot 3 \cdots m} = C_n^m$ є натуральне число.

Справді, помножуючи чисельник і знаменник на $(n-m)!$, отримується $C_n^m = \frac{n!}{m!(n-m)!}$; оскільки $n = m + (n-m)$, то внаслідок доведеної теореми C_n^m є натуральним числом.

1.3.2. Формули для числа дільників, суми дільників даного числа.

Особливо важливу роль у теорії чисел відіграють так званні мультиплікативні функції.

Функція $\theta(n)$ називається мультиплікативною, якщо: а) вона визначена для всіх натуральних n і не перетворюється в нуль хоч при одному такому значенні n; б) для довільних натурально взаємних простих n_1 і n_2 справедлива рівність:

$\theta(n_1 \cdot n_2) = \theta(n_1) \cdot \theta(n_2)$.

Приклад. Функція $\theta(n) = n^s$, де s - будь-яке дійсне, або комплексне число, є мультиплікативною. Справді, навіть при довільних n_1 і n_2 маємо:

$$\theta(n_1 \cdot n_2) = (n_1 n_2)^s = n_1^s \cdot n_2^s = \theta(n_1) \cdot \theta(n_2).$$

З означення мультиплікативної функції, зокрема, випливають такі її властивості:

1. $\theta(1) = 1$. Справді, якщо $\theta(n_0) \neq 0$, тоді

$\theta(n_0) = \theta(n_0 \cdot 1) = \theta(n_0)\theta(1)$.

Отже, $\theta(1) = 1$.

2. Якщо $\theta_1(n)$ *і* $\theta_2(n)$ - мультиплікативні функції, то їх добуток також буде мультиплікативною функцією.

Справді, позначаючи $\theta_0(n) = \theta_1(n) \cdot \theta_2(n)$, отримується наступне співвідношення:

$\theta_0(n) = \theta_1(n) \cdot \theta_2(n) = 1$;

а при $(n_1, n_2) = 1$ відповідно співвідношення:

$\theta_0(n_1 n_2) = \theta_1(n_1 n_2) \cdot \theta_2(n_1 n_2) = \theta_1(n_1)\theta_1(n_2)\theta_2(n_1)\theta_2(n_2) =$

$= \left[\theta_1(n_1)\theta_2(n_1)\right]\left[\theta_1(n_2)\theta_2(n_2)\right] = \theta_0(n_1)\theta_0(n_2),$

що й доводить твердження.

3. Якщо $\theta(n)$ - мультиплікативна функція, а $n_1, n_2, ... n_s$ - попарно взаємно прості числа, то $\theta(n_1, n_2, ... n_s) = \theta(n_1)\theta(n_2)...\theta(n_s)$.

Справді, для $s = 1$, 2 твердження справедливе; нехай воно справедливе для $s-1$ і доводиться його справедливість для s. Оскільки, $(n_i, n_j) = 1$, при всіх $i \neq j$, за умовою, то $(n_1, n_2, ..., n_{s-1}, n_s) = 1$. З означення мультиплікативної функції отримується: $\theta(n_1 n_2 ... n_{s-1}, n_s) = \theta(n_1 n_2 ... n_{s-1})\theta(n_s)$; але за припущенням $\theta(n_1 n_2 ... n_{s-1}) = \theta(n_1)\theta(n_2)...\theta(n_{s-1})$, і справедливість цієї властивості стає очевидною.

4. Нехай $\theta(n)$ - мультиплікативна функція і $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ - канонічний розклад числа n. Позначимо символом $\sum_{d/n}$ суму, поширену на всі натуральні дільники d числа n (включаючи 1 і саме n). При цих позначеннях справедлива така тотожність, яка виражає основну властивість мультиплікативних функцій:

$$\sum_{d/n} \theta(d) = \left[1 + \theta(p_1) + ... + \theta(p_1^{\alpha_1})\right] \cdots \left[1 + \theta(p_k) + ... + \theta(p_k^{\alpha_k})\right] \qquad (1.30)$$

(у випадку $n = 1$ треба вважати, що права частина дорівнює 1).

Для доведення цієї тотожності розкриємо дужки в її правій частині. Дістанемо суму доданків виду $\theta(p_1^{\beta_1}) \cdot \theta(p_2^{\beta_2}) \cdots \theta(p_k^{\beta_k})$, де $0 \le \beta_i \le \alpha_i (i = 1,2,...,k)$, або внаслідок мультиплікативності цієї функції, $\theta(p_1^{\beta_1} p_2^{\beta_2} ... p_k^{\beta_k}) = \theta(d)$, бо $p_1^{\beta_1} p_2^{\beta_2} ... p_k^{\beta_k}$ є не що інше, як дільники d числа n. З правила множення многочленна на многочлен випливає, що жоден такий доданок не буде пропущений і не повториться більше, ніж один раз. Тобто, отримується вираз, що стоїть в лівій частині тотожності (1.30).

При $\theta(n) = n^S$ тотожність (1) набере вигляду:

$$\sum_{d/n} d^S = (1 + p_1^S + p_1^{2S} + ... + p_1^{\alpha, s}) \cdots (1 + p_k^S + p_k^{2S} + ... + p_k^{\alpha, k^S}). \qquad (1.31)$$

Зокрема при $s = 1$ ліва частина тотожності (2) дає суму всіх натуральних дільників числа n; позначаючи її через $S(n)$, отримується вираз:

$$S(n)=\sum_{d/n} d=(1+p_1+p_1^2+...+p_1^{\alpha_1})\cdots(1+p_k+p_k^2+...+p_k^{\alpha_k}). \qquad (1.32)$$

Спрощуючи праву частину, отримується:

$$S(n)=\frac{p_1^{\alpha_1+1}-1}{p_1-1}\cdot\frac{p_2^{\alpha_2+1}-1}{p_2-1}\cdots\frac{p_k^{\alpha_k+1}-1}{p_k-1}. \qquad (1.33)$$

Вважаючи в тотожності (1.33) $s=1$, бачимо, що її ліва частина при цьому визначає число всіх натуральних дільників даного n, і позначаючи його через $\tau(n)$, отримується:

$$\tau(n)=(\alpha_1+1)(\alpha_2+1)...(\alpha_k+1). \qquad (1.34)$$

Зауважимо, що розкривши дужки в правій частині тотожності (1.32), отримуються всі дільники числа n.

Приклад. Знайти суму дільників, число дільників і самі дільники числа $680=2^3\cdot5\cdot17$.

$$S(680)=\frac{2^{3+1}-1}{2-1}\cdot\frac{5^{1+1}-1}{5-1}\cdot\frac{17^{1+1}-1}{17-1}=1620;$$

тоді:

$\tau(680)=(3+1)(1+1)(1+1)=16$.

Самі дільники числа 680 знаходяться, розкривши дужки у виразі $(1+2+4+8)(1+5)(1+17)$.

Значить:

1, 2, 4, 8, 5, 10, 20, 40, 17, 34, 68, 136, 85, 170, 340, 680 – всі дільники.

Функції $\tau(n)$ і $S(n)$ - мультиплікативні.

Справді, якщо $(a,b)=1$ і $a=p_1^{a_1}p_2^{a_2}...p_s^{a_s}$, і $b=q_1^{\beta_1},q_2^{\beta_2}...q_t^{\beta_t}$ - канонічні розклади чисел a і b, то $p_1^{a_1}p_2^{a_2}...p_s^{a_s}q_1^{\beta_1}q_2^{\beta_2}...q_t^{\beta_t}$ - канонічний розклад числа ab і тоді отримується наступне співвідношення:

$$S(ab)=\frac{p_1^{\alpha_1+1}-1}{p_1-1}...\frac{p_s^{\alpha_s+1}-1}{p_s-1}\cdot\frac{q_1^{\beta_1+1}-1}{q_1-1}...\frac{q_t^{\beta_t+1}-1}{q_t-1}=S(a)S(b);$$

$$\tau(ab)=(\alpha_1+1)...(\alpha_s+1)(\beta_1+1)...(\beta_t+1)=\tau(a)\tau(b).$$

Функції $S(n)$ і $\tau(n)$ є найпростішими прикладами мультиплікативних числових функцій; у них і аргумент, і значення функцій набувають тільки цілих додатних значень.

1.3.3. Функція Ейлера та її основні властивості.

Функція Ейлера $\varphi(n)$ визначається для всіх натуральних n і являє собою кількість натуральних чисел, менших від n і взаємно простих з n; при цьому припускається, що $\varphi(1)=1$.

Для невеликих значень n значення функції $\varphi(n)$ можна знайти простим підрахунком кількості чисел, менших від n і взаємно простих з n, наприклад, $\varphi(2)=1$, $\varphi(3)=2$, $\varphi(4)=2$, $\varphi(5)=4$, $\varphi(6)=2$, $\varphi(7)=6$, $\varphi(8)=4$, $\varphi(9)=6$ і т.д.

Визначимо значення $\varphi(n)$ для будь-якого натурального n.

Спочатку слід довести такі твердження.

Теорема. Функція Ейлера мультиплікативна, тобто $\varphi(m\cdot n)=\varphi(m)\cdot\varphi(n)$, якщо $(m,n)=1$. Щоб довести цю теорему, слід розмістити числа від 1 до mn у вигляді такої таблиці:

1,	2,…,	r,…,	m,
$m+1$,	$m+2$,…,	$m+r$,…,	$2m$,
$2m+1$,	$2m+2$,…,	$2m+r$,…,	$3m$,
……			
$(n-1)m+1$,	$(n-1)m+2$,	$(n-1)m+r$	$(n-1)m+m=mn$.

З даної таблиці визначаються кількість чисел, взаємно простих з mn. Взаємно простими з добутком mn будуть ті і тільки ті числа, які одночасно взаємно прості з m і n. Тому слід відібрати з таблиці спочатку всі числа, взаємно прості з m, а з них ті, які взаємно прості з n.

Числа одного стовпця або одночасно взаємні з m, або ні, бо $(r,m)=(m,km+r)$. Отже, можна говорити про «стовпці, взаємно простих з m» і визначити їх число за кількістю чисел, взаємно простих з m одного рядка, наприклад першого; тому кількість таких стовпців за означенням дорівнює $\varphi(m)$.

Тепер розглядається будь-який стовпець таблиці, наприклад:

$$r, m+r, 2m+r, ..., (n-1)m+r. \qquad (1.35)$$

Усього на цьому стовпці n чисел; слід показати, що всі вони при діленні на n даватимуть різні остачі. Справді, припускається супротивне, тобто:

$k_1m+r=nq_1+s$ і $k_2m+r=nq_2+s$,

де k_1, k_2 і s - цілі невід'ємні , менші від n. Тоді віднімаючи від першої рівності другу, отримується: $(k_1-k_2)m=n(q_1-q_2)$. Остання рівність показує, що $(k_1-k_2)m:n$, але $(m,n)=1$ за умовою, отже $(k_1-k_2):n$, але це неможливо, бо k_1 і k_2 різні і $|k_1-k_2|<n$. Отже, від ділення чисел ряду (2) таблиці на n отримуються остачі $s=0,1,2,3,...,n-1$; позначаючи через

$y = km + r = nq + s$ слід зауважити, що спільні дільники чисел y і n збігаються з спільними дільниками чисел n і s, зокрема, $(y, n) = (s, n)$. Отже, в ряді чисел (2) буде стільки взаємно простих з n, скільки їх буде в ряді 0, 1, 2, ..., n-1, тобто $\varphi(n)$. Отже, в таблиці є $\varphi(m) \cdot \varphi(n)$ чисел, взаємно простих як з m, так і з n, а отже, і з mn. З другого боку таблиця має всі числа від 1 до mn, і, отже, в ній $\varphi(m \cdot n)$ чисел, взаємно простих з mn, і ми дістанемо, що $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$ і теорему доведено.

Теорема. Нехай p – просте число і $a \geq 1$ - будь-яке натуральне число, тоді:

$$\varphi(p^a) = p^{a-1}(p-1) = p^a\left(1 - \frac{1}{p}\right). \tag{1.36}$$

Справді, розглянувши ряд чисел від 1 до p^a. Його можна подати в такому вигляді:

$1, 2, ..., p, ..., 2p, ..., 3p, ..., p \cdot p = p^2, ... p^{2-1} p = p^2$.

Зрозуміло, що цей ряд має p^{2-1} чисел, які діляться на p і, отже, не є взаємно простими з p^a; інші числа цього ряду не діляться на p, отже, вони будуть взаємно прості як з p, так і з p^a.

Отже, $\varphi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1) = p^a\left(1 - \frac{1}{p}\right)$.

Зокрема:

$$\varphi(p) = p - 1 \tag{1.37}$$

Теорема. Якщо $n > 1$ і $n = p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$ - канонічний розклад числа n, то:

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)...\left(1 - \frac{1}{p_k}\right). \tag{1.38}$$

Справді, внаслідок мультиплікативності, отримується вираз:

$$\varphi(n) = \varphi\left(p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}\right) = \varphi\left(p_1^{\alpha_1}\right)\varphi\left(p_2^{\alpha_2}\right)...\varphi\left(p_k^{\alpha_k}\right) =$$

$$= p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right) p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right)... p_k^{\alpha_k}\left(1 - \frac{1}{p_k}\right) =$$

$$= n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)...\left(1 - \frac{1}{p_k}\right).$$

Формулу (1.38) можна переписати наступним чином:

$$\varphi(n) = p_1^{\alpha_1-1}(p_1-1)p_2^{\alpha_2-1}(p_2-1)...p_k^{\alpha_k-1}(p_k-1). \qquad (1.39)$$

На практиці зручніше користуватись формулою (1.38).

Приклад. Знайти кількість чисел, менших за 1620 і взаємно простих з цим числом, тобто знайти $\varphi(1620)$. Маємо:

$$1620 = 2^2 \cdot 3^4 \cdot 5;\ \varphi(1620) = 1620\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{5}\right) = 432.$$

Теорема 4. Сума значень $\varphi(d)$, яка поширюється на всі наступальні дільники *d* числа *n,* дорівнює самому числу *n,* тобто:

$$\sum_{d/n} \varphi(d) = n. \qquad (1.40)$$

Справді, припускається, що $n = p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$ - канонічний розклад числа *n*. Через те, що функція Ейлера є мультиплікативною, формул (1.38) і (1.39) отримується:

$$\sum_{d/n} \varphi(d) = \left[1+\varphi(p_1)+\varphi(p_1^2)+...+\varphi(p_1^{\alpha_1})\right]...\left[1+\varphi(p_k)+\varphi(p_k^2)+...+\varphi(p_k^{\alpha_k})\right] =$$

$$= \left[1+(p_1-1)+p_1(p_1-1)+...+p_1^{\alpha_1-1}(p_1-1)\right]...\left[1+(p_k-1)+p_k(p_k-1)+...+p_k^{\alpha_k-1}(p_k-1)\right] =$$

$$= p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k} = n$$

Приклад. Перевірити тотожність (1.40) для *n=30.*

Дільники *d* числа 30 будуть: 1, 2, 3, 5, 6, 10, 15, 30;

$$\sum_{d/30} \varphi(d) = \varphi(1) + \varphi(2) + \varphi(3) + \varphi(5) + \varphi(6) + \varphi(10) +$$

$$+ \varphi(15) + \varphi(30) = 1 + 1 + 2 + 4 + 2 + 4 + 8 + 8 = 30.$$

1.3.4. Функція Мебіуса.

Функцією Мебіуса називається така числова функція $\mu(n)$, яка визначена для всіх натуральних *n* і характеризується наступними умовами:

1) $\mu(1) = 1$,

2) $\mu(n) = 0$, якщо *n* не ділиться на квадрат простого числа;

3) $\mu(n) = (-1)^k$, якщо *n* не ділиться на квадрат числа, відмінного від одиниці; при цьому *k* позначає число простих дільників *n*. Наприклад,

$\mu(2) = -1,\ \mu(3) = -1,\ \mu(4) = 0,\ \mu(5) = -1,\ \mu(6) = 1,\ \mu(63) = 0$ і т.д.

Отже, функція $\mu(n)$ набуває лише значення 0, 1 і -1. Неважко переконатися, що функція Мебіуса є також мультиплікативною функцією, тобто для будь-яких натуральних взаємно простих n_1 і n_2: $\mu(n_1 n_2) = \mu(n_1)\mu(n_2)$.

Справді, якщо хоч би одне з чисел n_1 або n_2 ділиться на квадрат простого числа, то очевидно $\mu(n_1 n_2) = 0$; $\mu(n_1)\mu(n_2) = 0$, тобто

$\mu(n_1 n_2) = \mu(n_1)\mu(n_2)$

припускається тепер, що

$n_1 = p_1 p_2 ... p_s, n_2 = q_1 q_2 ... q_t$,

де $p_1 p_2 ... p_s, n_2; q_1 q_2 ... q_t$ - різні прості числа, тоді

$\mu(n_1) = (-1)^s, \mu(n_2) = (-1)^t$

$\mu(n_1 n_2) = (-1)^{s+t} = \mu(n_1)\mu(n_2).$

Помноживши обидві частини цієї рівності на $\mu(d)$ і підсумувавши за всіма дільниками *d* числа *n*; тоді отримується:

$$\sum_{d/n} \mu(d) F\left(\frac{n}{d}\right) = \sum_{d/n} \sum_{\delta/\frac{n}{d}} \mu(d)\Phi(\delta).$$

Тут *d* і δ такі дільники числа *n*, що $\frac{n}{d\delta}$ - ціле число, тобто *d* можна вважати дільником числа $\frac{n}{\delta}$. Змінюючи порядок підсумовування в правій частині останньої рівності, отримується:

$$\sum_{\delta/\frac{n}{d}} \sum_{d/n} \mu(d)\Phi(\delta) = \sum_{\delta/\frac{n}{d}} \left[\Phi(\delta) \sum_{d/n} \mu(d) \right].$$

Але згідно з висновком 1, $\sum \mu(d) = 0$, крім випадку, коли $d = \frac{n}{\delta} = 1$, тобто коли $\delta = n$ він дорівнюватиме $\Phi(n)$. Звідси випливає, що:

$$\sum_{d/n} \mu(d) F\left(\frac{n}{d}\right) = \Phi(n). \qquad (1.41)$$

Формула (1.41) називається «формулою обернення» Дедекінда-Ліувілля. Вона записується так: $F(n) = \int \Phi(n)$, $\Phi(n) = DF(n)$ і $F(n)$ називається числовим інтегралом від $\Phi(n)$, взятим по дільниках, а $\Phi(n)$ називається числовою похідною від $F(n)$.

Приклад 1. Якщо $\Phi(n) = n$, то $F(n) = S(n)$. Це безпосередньо випливає з означення функції $S(n)$ і з рівності (1.41); аналогічно, якщо $\Phi(n) = 1$, то $F(n) = \tau(n)$.

Приклад 2. Якщо $F(n) = n = p_1^{a_1} \ldots p_n^{a_n} > 1$, то за формулою (1.41) отримується:

$$\Phi(n) = \sum_{d/n} \mu(d) \frac{n}{d} = n \sum_{d/n} \frac{\mu(d)}{d}.$$

Але $\sum_{d/n} \varphi(d) = n$, отже, $\varphi(n) = \Phi(n)$, бо $F(n) = n = \sum_{d/n} \Phi(d)$.

Використавши формулу (1.38), знаходиться значення функції Ейлера:

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\ldots\left(1 - \frac{1}{p_n}\right).$$

Отже, функцію Ейлера можна знайти іншим способом.

Важливе значення в теорії чисел має числова функція $\pi(x)$, яка визначає число простих чисел, що не перевищують дійсного числа x, наприклад, $\pi(1) = 0$, $\pi(2) = 1$, $\pi(3) = 2$, $\pi(\sqrt{7}) = 1$, $\pi\left(12\frac{1}{2}\right) = 5$ і т.д. Тут аргумент набуває довільних невід'ємних дійсних значень, а функція – лише цілих невід'ємних.

РОЗДІЛ 2

ПРОСТІ ЧИСЛА

2.1. Види простих чисел.

Просте число – це натуральне число, яке має рівно два натуральних дільника (лише 1 і саме число). Решту чисел, окрім одиниці, називають складеними. Таким чином, всі натуральні числа понад одиницю розбивають на прості і складені. Теорія чисел вивчає властивості простих чисел. В теорії кілець простим числам відповідають незвідні елементи.

Послідовність простих чисел починається так:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113 , 127, 131, 137, 139, 149, …

2.1.1. Розклад натуральних чисел на добуток простих.

Основна теорема арифметики стверджує, що кожне натуральне число більше одиниці (1), можна представити, як добуток простих чисел, причому, в єдиний спосіб з точністю до порядку множників. Таким чином, прості числа – це елементарні «будівельні блоки» натуральних чисел.

Представлення натурального числа у вигляді добутку простих називають розкладом на прості або факторизацією числа. На даний момент невідомі поліноміальні алгоритми факторизації чисел, хоча і не доведено, що таких алгоритмів не існує (мова йде про поліноміальні залежності часу роботи алгоритму від *логарифма* розміру числа, тобто від кількості його цифр). На припущенні про високу обчислювальну складність задачі факторизації базується криптосистема RSA.

2.1.2. Застосування простих чисел.

Великі прості числа (порядку 10^{300}) використовують в криптографії з відкритим ключем. Прості числа також використовують в хеш-таблицях і для генерації псевдовипадкових чисел (зокрема, в генераторі псевдовипадкових чисел Вихор Мерсена).

Ератосфен Киренський. Решето Ератосфена, решето Сундарама та решето Аткіна дають прості способи складання початкового списку простих чисел до певного значення.

Однак, на практиці, замість отримання списку простих чисел найчастіше потрібно перевірити, чи є дане число простим. Алгоритми,

які вирішують це завдання, називають тестами простоти. Існує безліч поліноміальних тестів простоти, але більшість з них є стохастичні (наприклад, тест Міллера-Рабина) і використовуються для потреб криптографії. Тільки в 2002 році було доведено, що завдання перевірки на простоту в загальному вигляді можна розв'язати за поліноміальний час, але запропонований детермінований алгоритм має досить велику складність, що ускладнює його застосування на практиці.

Для деяких класів чисел існують спеціалізовані ефективні тести простоти. Наприклад, для перевірки на простоту чисел Мерсена використовують тест Люка-Лемера, а для перевірки на простоту чисел Ферма – тест Пепіно.

2.1.3. Розподіл та кількість простих чисел.

Простих чисел нескінченно багато. Найдавніший відомий доказ цього факту було дано Евклідом в «Началах» (книга IX, твердження 20). Його доказ може бути коротко відтворено так: припускаємо, що кількість простих чисел скінченна. Перемножимо їх і додамо одиницю. Отримане число не ділиться ні на одне зі скінченного набору простих чисел, тому що залишок від ділення на будь-яке з них дає одиницю. Значить, добуток має ділитись на деяке просте число, не включене до цього набору.

Математики пропонували інші докази. Одне з них (наведене Ейлером) показує, що сума всіх чисел, зворотних до простих, розходиться.

Відома теорема про розподіл простих чисел стверджує, що кількість простих чисел менших за n, яке позначають як $\pi(n)$, росте як $n / \ln n$.

Теорема про розподіл простих чисел - теорема аналітичної теорії чисел, що описує асимптотику розподілу простих чисел. А саме, вона стверджує, що кількість $\pi(n)$ простих чисел на відрізку від 1 до n зростає із зростанням n як $n / \ln n$, тобто:

$$\frac{\pi(n)}{n / \ln n} \to 1,\ n \to \infty. \qquad (2.1)$$

Інакше кажучи, це означає, що у випадково вибраного числа від 1 до n, для достатньо великих n, ймовірність виявитися простим приблизно рівна $1 / \ln n$.

Також ця теорема може бути еквівалентним чином перефразована для опису поведінки k-го простого числа p_k: вона стверджує, що $p_k \sim k \ln k,\ k \to \infty$ (тут і далі запис $f \sim g$ означає $f/g \to 1$).

Ґрунтуючись на таблицях простих чисел, складених Фелкелем і Вегою, Лежандр припустив в 1796 році, що функція π(x) може бути

наближена виразом x / (ln$x$ − B), де B = 1.08... – константа, близька 1. Гаус, розглядаючи те ж питання і використовуючи доступні йому результати обчислень і деякі евристичні міркування розглянув іншу функцію – інтегральний логарифм $\mathrm{Li}(x) = \int_2^x \frac{1}{\ln x}\,dx$, проте не став публікувати цього твердження. Обидва наближення, як Лежандра, так і Гауса, приводять до однієї і тієї ж асимптотичної еквівалентності функцій π(x) і x / ln$x$, вказаної вище, хоча наближення Гауса і виявляється істотно кращим, якщо при оцінці помилки розглядати різницю функцій замість їх відношення.

У двох своїх роботах, 1848 і 1850 роки, Чебишев доводить, що верхня M і нижня m границі відношення:

$$\frac{\pi(x)}{x/\ln x}\ , \tag{2.2}$$

задовольняють нерівності $0.92129 \le m \le M \le 1.10555$, а також, що якщо границя відношення *(*)* існує, то вона рівна 1.

Загальний хід доведення.

Переформулювання в термінах псі-функції Чебишева.

Загальним початковим етапом міркувань є переформулювання твердження за допомогою псі-функції Чебишева, що визначається як:

$$\psi(x) = \sum_{p^k \le x} \log p\ , \tag{2.3}$$

іншими словами, псі-функція Чебишева це сума функції фон Мангольдта:

$$\psi(x) = \sum_{n \le x} \Lambda(n), \quad \Lambda(n) = \begin{cases} \log p,\ n = p^k, k \ge 1,\ p - \textit{просте} \\ 0,\ \textit{в інших випадках.} \end{cases} \tag{2.4}$$

А саме, виявляється, що асимптотичний закон розподілу простих чисел рівносильний тому, що $\psi(x) \sim x,\ x \to \infty$.

Це твердження є вірним тому, що логарифм «майже сталий» на більшій частині відрізка [1,n], а внесок квадратів, кубів, і т.д. в суму (2.4) є малим; тому практично всі логарифми ln$p$ приблизно рівні ln$x$, і функція ψ(x) асимптотично рівна π(x) *ln x*.

Класичні міркування: перехід до дзета-функції Рімана.

Як випливає з тотожності Ейлера $\zeta(s) = \prod_p \frac{1}{1} - p^{-s}$ ряд Діріхле, що відповідає функції фон Мангольдта, рівний мінус логарифмічній похідній дзета-функції:

$$\sum_n \Lambda(n) n^{-s} = -\frac{\zeta'(s)}{\zeta(s)}. \qquad (2.5)$$

Крім того, інтеграл по вертикальній прямій, що знаходиться праворуч від 0, від функції as / s рівний 2πi при a > 1 і 0 при 0 < a < 1. Тому, множення правої і лівої частини на $\frac{1}{2\pi i} x^s / s$ і інтегрування по вертикальній прямій по ds залишає в лівій частині суму Λ(n) з $n \leq x$. З іншого боку, застосування теореми про лишки дозволяє записати ліву частину у вигляді суми лишків; кожному нулю функції дзети відповідає полюс першого порядку її логарифмічної похідної, з лишком, рівним 1, а полюсу першого порядку в точці s = 1 – полюс першого порядку з лишком, рівним (− 1).

Строга реалізація цього викладу дозволяє одержати явну формулу Рімана:

$$\psi(x) = x - \sum_{\substack{\rho:\zeta(\rho)=0, \\ 0<\mathrm{Re}(\rho)<1}} \frac{x^\rho}{\rho} - \log(2\pi) - \frac{1}{2}\log(1 - x^{-2}), \qquad (2.6)$$

де сума обчислюється за нулями ρ дзета-функції.

Відсутність нетривіальних нулів дзета-функції поза критичною смугою спричиняє еквівалентність $\psi(x) \sim x$ (сума у формулі (2.6) зростатиме повільніше, ніж x).

Елементарне доведення: завершення Ердеша-Сельберга.

Основна теорема арифметики, що записується після логарифмування як:

$$\ln n = \sum_{p,k:p^k|n} \ln p, \qquad (2.7)$$

таким чином формулюється в термінах арифметичних функцій і згортки Діріхле як:

$$\ln = \Lambda \cdot 1, \qquad (2.8)$$

де ln і 1 – арифметичні функції, логарифм аргументу і тотожна одиниця відповідно.

Формула обертання Мебіуса дозволяє перенести 1 у праву частину:

$$\Lambda = \ln \cdot \mu, \qquad (2.9)$$

де μ – функція Мебіуса.

Сума лівої частини (2.9) – шукана функція ψ. У правій частині, застосування формули гіперболи Діріхле дозволяє звести суму згортки до суми:

$$\sum_k L(n/k)\mu(k), \tag{2.10}$$

де L – сума логарифма. Застосування формули Ейлера – Маклорена дозволяє записати $L(n)$ як:

$$L(n) = n\ln n - n + \frac{1}{2}\ln n + \gamma + o(1), \tag{2.11}$$

де γ – стала Ейлера. Виділяючи з цього виразу доданки, що мають вигляд:

$$\sum_k F(n/k), \tag{2.12}$$

для відповідним чином підібраної функції F (а саме F(x) = x − γ − 1), і позначаючи через R залишок, маємо через обертання Мебіуса:

$$\Lambda = F + \sum_k R(n/k)\mu(k). \tag{2.13}$$

Оскільки $F(x)\tilde{}x$, залишається перевірити, що другий доданок має вигляд $o(x)$. Застосування леми Аскера дозволяє звести цю задачу до перевірки твердження $M(x) = o(x)$, де $M(x) = \sum_{n\le x}\mu(n)$ – сума функції Мебіуса.

Малість сум функції Мебіуса на підпослідовності випливає з формули обертання, застосованої до функції 1 / n.

Далі, функція Мебіуса в алгебрі арифметичних функцій (з мультиплікативною операцією-згорткою) задовольняє «диференціальному рівнянню» першого порядку:

$$\mu' = -\mu \cdot \Lambda, \tag{2.14}$$

де $f'(n) = f(n)\cdot\ln n$ - диференціювання в цій алгебрі (перехід до рядів Діріхле перетворює його на звичайне диференціювання функції). Тому вона задовольняє і рівнянню другого порядку:

$$\mu'' = \mu \cdot (\Lambda \cdot \Lambda - \Lambda'). \tag{2.15}$$

Перехід до середнього у цьому рівнянні дозволяє те, що асимптотика суми функції $\Lambda 2 = \Lambda \cdot \Lambda + \Lambda$ оцінюється краще, ніж асимптотика сум Λ, дозволяє оцінювати відношення $M(x)/x$ через середні значення такого відношення. Така оцінка разом з «малістю за послідовністю» і дозволяє одержати шукану оцінку $M(x) = o(x)$.

2.1.4. Найбільше відоме просте число.

Здавна ведуться записи, в яких відзначають найбільші відомі на той час прості числа. Один з рекордів поставив свого часу Ейлер, знайшовши просте число $2^{31} - 1 = 2147483647$.

Найбільшим відомим простим числом станом на червень 2009 року є $2^{43112609} - 1$. Воно складається з 12 978 189 десяткових цифр і є простим числом Мерсена ($M_{43112609}$). Його знайшли 23 серпня 2008 року на математичному факультеті університету UCLA в рамках проекту по розподіленому пошуку простих чисел Мерсена GIMPS. Попереднє за величиною відоме просте, також є простим числом Мерсенна $M_{37156667}$, було знайдено 6 вересня 2007 року учасником проекту GIMPS Гансом-Міхаелем Елвеніхом (нім. *Hans-Michael Elvenich*).

Числа Мерсена вигідно відрізняються від решти наявністю ефективного тесту простоти: тесту Люка-Лемера. Завдяки йому прості числа Мерсена давно утримують рекорд, як найбільші відомі прості.

За знаходження простих чисел з понад 100 000 000 та 1 000 000 000 десяткових цифр EFF призначила грошові призи в 150 000 та 250 000 доларів США відповідно.

У 1859 році з'являється робота Рімана, що розглядає (введену Ейлером як функцію дійсного аргумента) ζ-функцію в комплексній області, і що пов'язує її поведінку з розподілом простих чисел. Розвиваючи ідеї цієї роботи, в 1896 році Адамар і Валле-Пуссен одночасно і незалежно доводять теорему про розподіл простих чисел.

Нарешті, в 1949 році з'являється доведення Ердеша-Сельберга, що не використовує понять комплексного аналізу.

2.2. Властивості простих чисел.

Якщо p – просте, і p ділить ab, то p ділить a або b. Цю властивість довів Евкліда, і відома вона, як лема Евкліда. Її використовують при доведенні основної теореми арифметики.

Кільце остач Z_n є полем тоді і тільки тоді, коли n – просте.

Характеристика кожного поля – нуль або просте число.

Якщо p – просте, a − натуральне, то $ap - a$ ділиться на p (мала теорема Ферма).

Якщо G – скінченна група з pn елементів, то G містить елемент порядку p.

Якщо G – скінченна група, і pn – максимальний ступінь p , який ділить $|G|$, то G має підгрупу порядку pn, яку називають підгрупою Силова,

більше того, кількість підгруп Силова дорівнює $pk + 1$ для деякого цілого k (теореми Силова).

Натуральне $p > 1$ є простим тоді і тільки тоді, коли $(p - 1)! + 1$ ділиться на p (теорема Вільсона).

Якщо $n > 1$ – натуральне, то існує просте p , Таке, що $n < p < 2n$ (постулат Бертрана).

Ряд чисел, зворотних до простих, розходиться. Більш того, при $x \to \infty$

$$\sum_{p<x} \frac{1}{p} \approx \ln \ln x.$$

Будь-яка арифметична прогресія виду $a,a + q,a + 2q,a + 3q,...,$ Де $a,q>1$ – цілі взаємно-прості числа, містить нескінченно багато простих чисел (Теорема Діріхле про прості числах в арифметичній прогресії).

Будь-яке просте число більше 3, можна представити у вигляді $6k + 1$, або у вигляді $6k - 1$, де k – деяке натуральне число.

Якщо $p > 3$ – просте, то $p2 - 1$ кратне 24.

Множина додатних значень многочлена

$(k + 2)(1 - [wz + h + j - q]2 - [(gk + 2g + k + 1)(h + j) + h - z]2 - [2n + p + q + z - e]2 -[16(k + 1)3(k + 2)(n + 1)2 + 1 - f2]2 - [e3(e + 2)(a + 1)2 + 1 - o2]2 - [(a2 - 1)y2 + 1 - x2]2 -[16r2y4(a2 - 1) + 1 - u2]2 - [((a + u2(u2 - a))2 - 1)(n + 4dy)2 + 1 - (x + cu)2]2 - [n + l + v - y]2 -[(a2 - 1)l2 + 1 - m2]2 - [ai + k + 1 - l - i]2 - [p + l(a - n - 1) + b(2an + 2a - n2 - 2n - 2) - m]2 -[q + y(a - p - 1) + s(2ap + 2a - p2 - 2p - 2) - x]2 - [z + pl(a - p) + t(2ap - p2 - 1) - pm]2)$

при невід'ємних цілих значеннях змінних збігається з множиною простих чисел. Даний результат є окремим випадком доведеною Юрієм Матіясевічем діофантності будь-якої ефективно зліченної множини.

Досі існує багато відкритих запитань відносно простих чисел, найвідоміші з яких були перераховані Едмундом Ландау на п'ятому Міжнародному математичному конгресі.

Проблема Гольдбаха (перша проблема Ландау): довести або спростувати, що кожне парне число, більше двох, може бути представлено у вигляді суми двох простих чисел, а кожне непарне число, більше 5, може бути представлено у вигляді суми трьох простих чисел.

Друга проблема Ландау : чи нескінченна множина «простих близнюків» – простих чисел, різниця між якими дорівнює 2?

Гіпотеза Лежандра (третя проблема Ландау) чи вірно, що між n2 і $(n + +1)2$ завжди знайдеться просте число?

Четверта проблема Ландау: чи нескінченна множина простих чисел виду $n2 + 1?$

Відкритої проблемою є також існування нескінченної кількості простих чисел у багатьох цілочисельних послідовностях, включаючи числа Фібоначчі, числа Ферма і т. д.

2.2.1 Види простих чисел.

1. Число Мерсенна.

Число Мерсена (Mersenne number) – числа виду $M_n = 2n - 1$, де n – натуральне число. Числа називають іменем французького математика Марена Мерсенна, що жив на початку XVII століття.

Послідовність чисел Мерсенна починається так:

1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, …

Іноді числами Мерсенна називають числа Mp з простими індексами p. Ця послідовність починається так:

3, 7, 31, 127, 2047, 8191, 131071, 524287, 8388607,…

Основні властивості.

Будь-який дільник числа Mp для простого p має вигляд 2pk + 1, де k – ціле число. (Це прямий наслідок малої теореми Ферма)

Ейлер довів, що кожне парне досконале число має вигляд 2p − 1Mp, де число Мерсенна Mp є простим.

Прості числа Мерсенна.

Числа Мерсенна є добре відомими в зв'язку з ефективним критерієм простоти Люка-Лемера, завдяки якому прості числа Мерсенна давно утримують лідерство як найвідоміші прості числа. На даний час найбільшим відомим простим числом є число Мерсенна $M_{32582657}$ = 232582657 − 1, знайдене в вересні 2006 року в рамках проекту розподілених обчислень GIMPS. Всього відомо 44 простих числа Мерсенна, при чому порядкові номери встановлені лише у перших 39-ти (точно).

Послідовність простих чисел Мерсенна і їх показників починається так:

Mp: 3, 7, 31, 127, 8191, 131071, 524287, 2147483647, 2305843009213693951, 618970019642690137449562111, …

Відкриті проблеми простих чисел Мерсена.

Нескінченність кількості простих чисел Мерсенна і їх асимптотика.

Простота числа

$M_{M_{61}} = 2^{2^{61}-1} - 1$.

2. Прості числа Белла

Прості числа, які є числом розбиття множини з n елементами.

2, 5, 877, 27644437, 35742549198872617291353508656626642567, 359334085968622831041960188598043661065388726959079837. Наступне число має 6539 цифр.

3. Кубічні прості числа.

Прості числа вигляду $\frac{x^3 - y^3}{x - y}, x = y + 1$

7, 19, 37, 61, 127, 271, 331, 397, 547, 631, 919, 1657, 1801, 1951, 2269, 2437, 2791, 3169, 3571, 4219, 4447, 5167, 5419, 6211, 7057, 7351, 8269, 9241, 10267, 11719, 12097, 13267, 13669, 16651, 19441, 19927, 22447, 23497, 24571, 25117, 26227, 27361, 33391, 35317,

а також $\frac{x^3 - y^3}{x - y}, x = y + 2$

13, 109, 193, 433, 769, 1201, 1453, 2029, 3469, 3889, 4801, 10093, 12289, 13873, 18253, 20173, 21169, 22189, 28813, 37633, 43201, 47629, 60493, 63949, 65713, 69313, 73009, 76801, 84673, 106033, 108301, 112909, 115249.

4. Прості-близнюки.

(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139), (149, 151), (179, 181), (191, 193), (197, 199), (227, 229), (239, 241), (269, 271), (281, 283), (311, 313), (347, 349), (419, 421), (431, 433), (461, 463), (521, 523), (569, 571), (599, 601), (617, 619), (641, 643), (659, 661), (809, 811), (821, 823), (827, 829), (857, 859), (881, 883).

5. Прості, які складаються з одиниць.

Числа, Які складаються 2, 19, 23, 317, 1031, 49081, 86453, 109297, 270343 одиниць, є простими.

6. Прості, які складаються з одиниць та нулів.

Крім простих чисел, які складаються тільки з одиниць, можна відмітити і прості числа, які складаються з одиниць та нулів. В межах перших десяти мільйонів простими є наступні з таких чисел:

11, 101, 10111, 101111, 1011001, 1100101...

7. Прості числа Вільсона.

Прості числа p, для яких $(p - 1)! + 1$ ділиться націло на p^2.

Відомі прості Вільсона: 5, 13, 563.

Інші прості Вільсона невідомі. Гарантовано не існує інших простих Вільсона менших 500 000 000.

8. Прості числа Вольстенхольма.

Прості числа p для яких біноміальний коефіцієнт.

$$\binom{2p-1}{p-1} \equiv 1 (\mathrm{mod}\, p^4)$$

Відомі тільки ці числа до мільярда: 16843, 2124679

9. Прості числа Керола.

Прості числа вигляду $(2^n - 1)^2 - 2$.

7, 47, 223, 3967, 16127, 1046527, 16769023, 1073676287, 68718952447, 274876858367, 4398042316799, 1125899839733759, 18014398241046527, 1298074214633706835075030044377087.

10. Прості числа Маркова.

Прості числа p для яких існують цілі x і y такі, що $x^2 + y^2 + p^2 = 3xyp$.

2, 5, 13, 29, 89, 233, 433, 1597, 2897, 5741, 7561, 28657, 33461, 43261, 96557, 426389, 514229.

11. Прості числа Мерсена.

Прості числа вигляду $2^n - 1$. Перші 12 чисел:

3, 7, 31, 127, 8191, 131071, 524287, 2147483647, 2305843009213693951, 618970019642690137449562111, 162259276829213363391578010288127, 170141183460469231731687303715884105727

На сьогоднішній день відомо 46 простих чисел Мерсена, саме більше відоме просте число є числом Мерсена.

12. Прості числа Ньюмана-Шенкса-Вільямса.

Числа Ньюмана-Шенкса-Вільямса є простими.

7, 41, 239, 9369319, 63018038201, 489133282872437279, 1917500294268803292859 9.

13. Прості числа Прота.

Прості числа вигляду $P=k\cdot 2^n+1$, причому k непарне і $2^n > k$.

14. Прості числа Софі Жермен.

Прості числа p такі, що $2p + 1$ також прості.

2, 3, 5, 11, 23, 29, 41, 53, 83, 89, 113, 131, 173, 179, 191, 233, 239, 251, 281, 293, 359, 419, 431, 443, 491, 509, 593, 641, 653, 659, 683, 719, 743, 761, 809, 911, 953.

15. Прості числа Ферма.

Це прості числа вигляду $2^{2^n}+1$.

Відомі числа Ферма: 3, 5, 17, 257, 65537.

16. Прості числа Чена.

Такі прості числа p, що $p + 2$ або просте, або напівпросте.

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 47, 53, 59, 67, 71, 83, 89, 101, 107, 109, 113, 127, 131, 137, 139, 149, 157, 167, 179, 181, 191, 197, 199, 211, 227, 233, 239, 251, 257, 263, 269, 281, 293, 307, 311, 317, 337, 347, 353, 359, 379, 389, 401, 409.

17. Збалансовані прості числа.

Прості числа, які є середнім арифметичним попереднього простого числа и наступного простого числа.

5, 53, 157, 173, 211, 257, 263, 373, 563, 593, 607, 653, 733, 947, 977, 1103, 1123, 1187, 1223, 1367, 1511, 1747, 1753, 1907, 2287, 2417, 2677, 2903,

2963, 3307, 3313, 3637, 3733, 4013, 4409, 4457, 4597, 4657, 4691, 4993, 5107, 5113, 5303, 5387, 5393.

18. Унікальні прості числа.

Прості числа *p*, довжина періодичного дробу яких від $1/p$ унікальна.

3, 11, 37, 101, 9091, 9901, 333667, 909091, 99990001, 999999000001, 9999999900000001, 909090909090909091, 1111111111111111111, 11111111111111111111111, 900900900900990990990991.

19. Факторіальні прості.

Це прості числа виду n!±1 для деякого $n \in N$:

2, 3, 5, 7, 23, 719, 5039, 39916801, 479001599, 87178291199, 10888869450418352160768000001, 265252859812191058636308479999999, 263130836933693530167218012159999999, 86833176188118864955181944012799999999.

20. Центральні квадратні прості числа.

Числа виду $n^2 + (n + 1)^2$.

5, 13, 41, 61, 113, 181, 313, 421, 613, 761, 1013, 1201, 1301, 1741, 1861, 2113, 2381, 2521, 3121, 3613, 4513, 5101, 7321, 8581, 9661, 9941, 10513, 12641, 13613, 14281, 14621, 15313, 16381, 19013, 19801, 20201, 21013, 21841, 23981, 24421, 26681.

21. Центральні трикутні прості числа.

Числа вида $(3n^2 + 3n + 2) / 2$.

19, 31, 109, 199, 409, 571, 631, 829, 1489, 1999, 2341, 2971, 3529, 4621, 4789, 7039, 7669, 8779, 9721, 10459, 10711, 13681, 14851, 16069, 16381, 17659, 20011, 20359, 23251, 25939, 27541, 29191, 29611, 31321, 34429, 36739, 40099, 40591, 42589.

22. Центральні семигранні прості числа.

Числа виду $(7n^2 - 7n + 2) / 2$.

43, 71, 197, 463, 547, 953, 1471, 1933, 2647, 2843, 3697, 4663, 5741, 8233, 9283, 10781, 11173, 12391, 14561, 18397, 20483, 29303, 29947, 34651, 37493, 41203, 46691, 50821, 54251, 56897, 57793, 65213, 68111, 72073, 76147, 84631, 89041, 93563.

23. Центральні десятигранні прості числа.

Числа виду $5(n^2 - n) + 1$.

11, 31, 61, 101, 151, 211, 281, 661, 911, 1051, 1201, 1361, 1531, 1901, 2311, 2531, 3001, 3251, 3511, 4651, 5281, 6301, 6661, 7411, 9461, 9901, 12251, 13781, 14851, 15401, 18301, 18911, 19531, 20161, 22111, 24151, 24851, 25561, 27011, 27751.

2.3. Тести перевірки простих чисел.

Тест простоти – алгоритм перевірки, чи є дане число простим. Важливо наголосити на різниці між тестуванням простоти та факторизацією цілих чисел. Станом на 2009 рік, факторизація є обчислювально важкою проблемою, в той час як тестування простоти є порівняно простішим (має поліноміальну складність).

Решето Ератосфе́на в математиці – простий стародавній алгоритм знаходження всіх простих чисел менших деякого цілого числа *n*, що був створений давньогрецьким математиком Ератосфеном. Він є попередником сучасного решета Аткіна, швидшого, але і складнішого алгоритму.

Метод

Якщо потрібно знайти всі прості числа менші за певне число *N*, виписуються всі числа від *1* до N^2 *-1*. Потім в цьому ряду викреслюються всі числа, які діляться на 2,3, 4 і так далі до N. Числа, які залишилися не викресленими після цієї процедури - прості.

Приклад для *n* = 20.

Запишемо натуральні числа від 2 до 20 в рядок:

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Перше число в рядку 2 — просте. Викреслимо всі числа кратні 2 (кожне друге, починаючи з $2^2 = 4$):

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~

Наступне невикреслене число 3 — просте. Викреслимо всі числа кратні 3 (кожне третє, починаючи з $3^2 = 9$):

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~

Наступне не викреслене число 5 — просте. Викреслимо всі числа кратні 5 (кожне п’яте, починаючи з $5^2 = 25$). І т. д.

Необхідно викреслити кратні для всіх простих чисел *p*, для яких $p^2 \leq n$. В результаті всі складені числа будуть викреслені, а залишаться всі прості числа. Для *n = 20* вже після викреслювання кратних числу 3 всі складені числа виявляються викресленими.

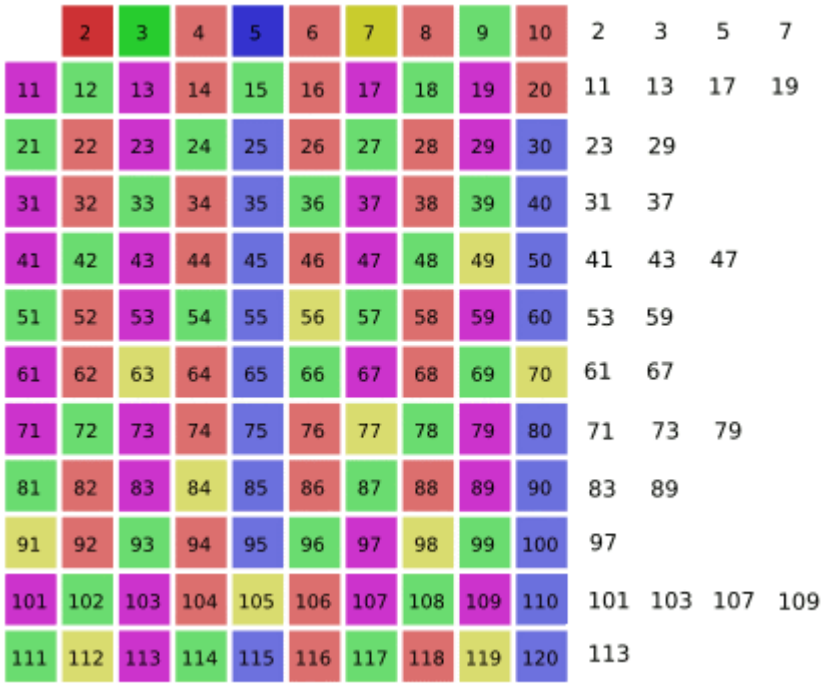
Алгоритм

1. Записується список чисел від 2 до найбільшого, про яке потрібно дізнатися чи є простим. Позначається: *Список А*.
2. Записується число 2, перше просте число, в інший список для знайдених простих чисел. Позначається: *Список В*.
3. Викреслюються 2 і всі кратні 2 числа зі Списку А.
4. Перше (найменше) не викреслене число в Списку А є простим. Записується його в Список В.
5. Викреслюється це число і всі кратні йому числа зі Списку А. Викреслювання кратних можна почати з числа, яке є квадратом поточного

простого числа, бо менші кратні були викреслені на попередньому кроці (наприклад, 6 було викреслене як 2*3 і викреслювати його як 3*2 вже не треба, тобто починаємо з $3*3=3^2$).

6. Повторюються кроки 4 і 5 до тих пір, поки в Списку А не залишиться чисел.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120



Решето Сундарама

В математиці решето Сундара́ма – детермінований алгоритм знаходження всіх простих чисел до деякого цілого числа *n*. Алгоритм було розроблено індійським студентом С. П. Сундарамом в 1934 році.

Формалізація алгоритму

Із ряду чисел від 1 до *N* виключаються всі числа, що мають вид $Z=i+j+2ij$,

де $i=1,2,3,\ldots,n$; $j=1,2,3,\ldots,i$, а кожне із чисел, що залишилися, помножується на 2 і до нього додається 1. Послідовність, що виникає таким чином, є послідовністю непарних простих чисел.

Кількість обчислень можна дещо зменшити, якщо відзначити наступне: в разі i>N/3 Z виходить за межі N вже при j=1, і, відповідно, можна зменшити діапазон значень змінної і.

Складність цього алгоритму складає $\Theta(N\ln N)$, що гірше, ніж у решета Ератосфена $\Theta(N\ln\ln N)$.

Найпростіший тест простоти полягає в такому: коли задане число *n*, перевірити чи якесь ціле *m* від 2 до *n-1* ділить *n*. Якщо *n* ділиться на певне *m*, то *n* складене, в іншому разі воно просте. Замість перевірки всіх *m* до *n-1*, досить лише перевірити *m* до $\sqrt{n}$: якщо *n* складене, то його можна розкласти на два множники, принаймні один з яких не перевищує $\sqrt{n}$. Можна також покращити ефективність, пропускаючи всі парні *m* , за

винятком 2, бо коли якесь парне число ділить n , то 2 також ділить. Можна далі вдосконалити зауважуючи, що всі прості числа, за винятком 2 та 3, мають вигляд $6k \pm 1$. Дійсно, всі цілі можна подати як $(6k + i)$ для деякого k та для $i = -1, 0, 1, 2, 3$, або 4; 2 ділить $(6k + 0), (6k + 2), (6k + 4)$; а 3 ділить $(6k + 3)$. Спочатку перевіряємо чи n ділиться на 2 або 3, тоді пробігаємо всі числа вигляду $6k \pm 1 \leq \sqrt{n}$. Це у 3 рази швидше від попереднього методу.

Спостереження, аналогічні до попереднього, можна застосувати рекурсивно, отримуючи решето Ератосфена. Вдалим способом пришвидшення цих методів (і всіх інших згаданих далі) є попередній обрахунок і зберігання списку всіх простих до певної межі, скажімо всіх простих до 200. (Такий список можна обчислити за допомогою решета Ератосфена). Тоді, перед тестуванням n на простоту з використанням серйозного методу, спочатку перевіряємо чи n не ділиться на якесь просте із цього списку.

Ймовірнісні тести

Найбільш популярними тестами простоти є ймовірнісні тести. Ці тести використовують, крім тестованого числа n, деякі інші числа a , які випадково вибираються з певного набору; звичні рандомізовані тести простоти ніколи не оголошують прості числа складеними, але можливе для складених чисел оголошення їх простими. Імовірність помилки можна зменшити, повторюючи тест з різними незалежно вибраними a; для двох найчастіше вживаних тестів, для *будь-якого* складеного n принаймні половина a визначає складеність n , тому k повторень зменшують імовірність помилки до щонайбільше 2^{-k}. Останню величину можна зробити як завгодно малою, збільшуючи k.

Базова структура рандомізованих тестів простоти є такою:

1. Випадково вибрати число a.
2. Перевірити певну рівність, що містить a та задане число n. Якщо рівність не виконується, то n є складене число, a називають *свідченням* складеності, і тест зупиняється.
3. Виконувати крок 1, поки не буде досягнуто потрібної певності.

Після низки повторень, якщо не отримано, що n є складене число, то його можна оголосити імовірнісним простим.

Найпростішим ймовірнісним тестом простоти є тест простоти Ферма. Це лише евристичний тест; деякі складені числа (числа Кармайкла) будуть оголошені "ймовірнісними простими" незалежно від того, яке свідчення вибране. Проте, він деколи використовується з метою швидкої перевірки числа, наприклад, на фазі утворення ключа криптографічного алгоритму з відкритим ключем RSA.

Тест простоти Міллера-Рабіна та Тест простоти Соловея-Штрассена є вдосконаленими варіантами, які визначають всі складені числа (це означає:

для *кожного* складеного числа *n*, принаймні 3/4 (Міллер-Рабін) або 1/2 (Соловей-Штрассен) чисел *a* є свідченнями складеності *n*). На ці методи часто падає вибір, бо вони набагато швидші, ніж інші загальні тести простоти.

Леонард Адлеман та Хуанг запропонували варіант без помилки (але лише з очікуваним поліноміальним часом виконання) тесту простоти на основі еліптичних кривих. На відміну від інших імовірнісних тестів, цей алгоритм дає сертифікат простоти, а тому може бути використаний для доведення простоти числа. Цей алгоритм занадто повільний на практиці.

Швидкі детерміновані тести

Першим детермінованим тестом простоти значно швидшим, ніж наївні методи, був циклотомічний тест; для часу його виконання отримано оцінку $O((\log n)^{c\log(\log(\log(n)))})$, де *n* тестоване на простоту число, а *c* константа, незалежна від *n*. Це повільніше, ніж поліноміальний час.

Для тесту простоти на основі еліптичних кривих можна отримати оцінку $O((\log n)^6)$, але лише коли використовуємо деякі ще не доведені (але які як правило припускаються вірними) положення аналітичної теорії чисел. Це один з з найчастіше вживаних на практиці детермінованих тестів.

Реалізація цих двох методів досить важка, бо є великий ризик помилок при програмуванні; це одна з причин, чому їм не віддають перевагу.

Якщо вважається вірною узагальнена гіпотеза Рімана, то тест Міллера-Рабіна можна звести до детермінованої версії з часом виконання $O((\log n)^4)$. На практиці, цей алгоритм повільніший, ніж два інших для величин чисел, з якими можна реально оперувати.

У 2002, Маніндра Агравал, Нітін Саксена та Нірай Кайал описали новий детермінований тест простоти, AKS тест простоти, який як доведено виконується за $O((\log n)^{12})$. Крім того, якщо вірна гіпотеза Харді-Літлвуда, яку вважають справедливою, то він виконується за $O((\log n)^6)$. Отже, маємо перший детермінований тест простоти з доведеним поліноміальним часом виконання. На практиці, цей алгоритм повільніший, ніж імовірнісні методи.

Складність

У теорії складності обчислень, формальну мову, яка відповідає простим числам, позначають PRIMES. Неважко показати, що PRIMES належить до **Co-NP**: її доповнення COMPOSITES належить до **NP**, бо можна показати складеність недетерміновано вгадуючи дільник.

У 1975 Вауган Пратт показав існування сертифікату простоти, який перевіряється за поліноміальний час, і значить PRIMES належить до NP, а тому й до NP ? CoNP. Деталі дивись у сертифікат простоти.

Подальше відкриття алгоритмів Соловея-Штрассена та Міллера-Рабіна показало належність PRIMES до CoRP. У 1992 алгоритм Адлемана-

Хуанга звузив складність до ZPP = RP ? coRP, що є заміщенням результату Пратта.

Циклотомічний тест Адлемана, Померанце та Рамлі 1983 р. показав належність PRIMES до QP (квазі-поліноміальний час), для якого невідоме порівняння із згаданими раніше класами.

Існування AKS тесту простоти, який остаточно розв'язав цю давню проблему, означає, що PRIMES належить до P.

Теоретико-числові методи

Існують певні теоретико-числові методи для тестування чи є число простим, зокрема тест Лукаса-Лемера та тест Профа. Як правило, для цих тестів потрібний розклад $n + 1$, $n - 1$, або аналогічних чисел, а це означає, що вони не підходять для тестування простоти чисел загального вигляду, проте часто є досить потужним засобом, коли тестуємо число n спеціального вигляду.

Тест Лукаса-Лемера спирається на факт, що мультиплікативний порядок числа a за модулем n дорівнює $n - 1$ для простого n, якщо a примітивний корінь за модулем n. Коли можемо показати, що a примітивний корінь для n, то можемо довести простоту n.

РОЗДІЛ 3

ПОНЯТТЯ АЛГЕБРАЇЧНИХ СТРУКТУР. ГРУПИ

3.1. Алгебраїчні структури.

Бінарна операція виконується над парами елементів певної множини. Якщо операція виконуються над одним об'єктом, то її називають унітарною.

Означення. Нехай *M* – множина елементів a, b, c, ... довільної природи. Тоді в множині *M* - введено деяку унітарну операцію, яка кожному елементу a множини *M* поставлено у відповідність єдиний елемент b цієї ж множини.

Унітарна операція в множині *M* є, таким чином, однозначним відображенням множини *M* самої в себе. Прикладами унітарної операції є операція піднесення числа до квадрата, взяття доповнення підмножини даної множини, заперечення висловлення, знаходження абсолютної величини дійсного числа і ін.

В алгебрі розглядаються також *n*-арні операції, тобто операції, що виконуються над упорядкованими системами *n* елементів даної множини.

Означені вище унітарні й бінарні операції виконуються над елементами певної множини, і результат виконання операції належить також до тієї ж множини, тому їх називають внутрішніми операціями або внутрішніми законами композиції. Поряд з внутрішніми законами композиції в алгебрі розглядаються також зовнішні закони композиції, в яких, крім основної множини *M*, бере участь ще й допоміжна множина *Q,* елементи якої називають операторами. Зовнішній закон композиції парі $<\alpha, a>$, утвореної оператором α і елементом *a*, ставить у відповідність деякий цілком визначений елемент b множини *M*. Точне означення зовнішнього закону композиції формулюють так.

Означення. Зовнішнім законом композиції елементів множини Q, що називається множиною операторів закону, і елементів множини M називають правило, за яким кожній парі (α, a) ($\alpha \in$ Q, a. $\in$M) ставиться у відповідність єдиний елемент b множини *M*, тобто відображення ψ : Q x M $\rightarrow$M.

Композицію елементів α і *a* при цьому позначають символом α• , або α*a*. і називають добутком елемента *a* на оператор α.

Множення всіх елементів множини *M* на деякий оператор α *Є Q* є, очевидно, унітарна операція в множині *M*. Отже, завдання зовнішнього закону композиції елементів з *Q* і елементів з *M* рівносильне введенню в множині M певної кількості унітарних операцій. Як приклад зовнішнього закону композиції можна назвати множення вектора площини на дійсне число.

Означення. Множина *M,* для елементів якої задано один або кілька законів композиції, називається алгебраїчною структурою.

Завдання алгебри є вивчення алгебраїчних структур, однак, алгебра вивчає далеко не всі алгебраїчні структури. Можна побудувати чимало прикладів алгебраїчних структур, але в переважній більшості вони не матимуть ніяких застосувань ні в теорії, ні в практиці, а «теорія» таких структур складатиметься з означень і тривіальних наслідків з них. Такі структури, очевидно, не можуть бути об'єктом вивчення.

У процесі розвитку математики виділилася й стала докладно вивчатися невелика кількість основних типів алгебраїчних структур, алгебраїчні операції в яких за своїми властивостями більш-менш близькі до операцій додавання і множення чисел. Найважливішими серед різних алгебраїчних структур є група, кільце, поле, лінійний простір, лінійна алгебра. Вивчення властивостей саме цих алгебраїчних структур, опис їх будови і зв'язків між ними й іншими основними математичними об'єктами є одним з найважливіших завдань алгебри теорій цифрових автоматів, систем захисту інформаційних потоків на сучасному етапі її розвитку.

3.2. Групи, основні поняття.

3.2.1. Формальне визначення групи.

Групою називається множина певних елементів, якщо на ній виконуються певні аксіоми.

Аксіома I. Закон композиції елементів, коли кожним двом елементам *A, B*, взятим в певному порядку, можна співставити третій елемент *C* тієї ж сукупності. Записуємо це співставлення як звичайне множення: $AB = C$.

Аксіома II. Має місце асоціативний (сполучний) закон: $(AB)\,C = A\,(BC)$.

Аксіома III. Існує права одиниця *J*, тобто такий елемент, який не змінює будь-якого елемента *X* множини, якщо на нього помножити цей елемент праворуч: $XJ = X$.

Аксіома IV. Для кожного елементу *A* сукупності можна знайти його правий обернений елемент *X*, від множення на який праворуч елемент *A* перетворюється в *J*, тобто $AX = J$.

З-поміж цих аксіом аксіома I є, безумовно, необхідною взагалі для будь-якої теорії композиції символів. Аксіома II характерна для сукупностей перетворень. Під перетворенням розуміють перехід від елементів певної множини $\mathfrak{M}$ до інших елементів тієї самої множини.

Приклад 1. Множина складається з скінченого числа предметів, наприклад з x_1, x_2, x_1. Перетворення полягає в тому, що з x_1 зіставляється x_2, (будемо надалі говорити: x_1 переходить в x_2), $x_2 \rightarrow x_3$, $x_3 \rightarrow x_1$.Такого роду

перетворення носить назву підстановки записується так: $x_1 \to x_2$, $x_2 \to x_3$, $x_3 \to x_1$, або в матричному вигляді: $\begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_1 \end{pmatrix}$.

Приклад. Множина складається з усіх цілих (додатних та від'ємних) чисел. Перетворення полягає у додаванні до кожного елемента множини числа 1. Це перетворення можна записати так: $x \to x+1$.

Приклад. Множина складається з усіх дійсних значень змінної x.

Перетворення полягає в зіставленні з кожним значенням x значення $\frac{ax+b}{cx+d}$, де a і b – сталі числа. Таке перетворення називається дробовою лінійної підстановкою і позначається так: $x \to \frac{ax+b}{cx+d}$.

Для того, щоб ввести поняття групи перетворень, необхідно ввести поняття композиції (або множення) двох перетворень. Якщо перетворення A переводить елемент M множини $\mathfrak{M}$ в M', а перетворення B переводить M' в M'', то під AB ми будемо розуміти таке перетворення, яке переводить M в M''. Змушуючи елемент M пробігати всю множину $\mathfrak{M}$, ми тим самим цілком визначимо перетворення AB.

Приклад. Нехай $A = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_1 \end{pmatrix}$, $B = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_1 & x_3 & x_2 \end{pmatrix}$. Якщо A переводить x_1 в x_2, а $B – x_2$ в x_3, то AB має переводити x_1 в x_3. Розмірковуючи таким чином щодо «кожного з елементів x_1, x_2, x_3, отримаємо: $AB = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_3 & x_2 & x_1 \end{pmatrix}$.

Відзначимо, що результат множення залежить від порядку, в якому розміщені множники A, B. У нашому прикладі ми бачимо, що $A = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_1 & x_3 \end{pmatrix}$, тобто що $AB \neq BA$. У цьому випадку говорять, що група не підкоряється комутативному закону.

Приклад. Нехай перетворення A переводить змінну x в $x + a$, а перетворення B переводить x в $x + b$. Тоді AB переводить x в $(x+b) + a = x + (a + b)$. Тут $AB = BA$, а тому група називається комутативною або абелевою, на ім'я математика Абеля (N.H. Abel), що вперше вивчив рівняння з комутативними групами.

Для всіх сукупностей перетворень має місце асоціативний закон. Нехай дано перетворення A, B, C, і A переводить елемент M множини $\mathfrak{M}$ в M', B переводить M' в M'' і C переводить M'' в M'''. Тоді видно, що як

(AB) C, так і *A (BC)* переводить *M* в M'''. Але оскільки в якості *M* ми можемо взяти будь-який елемент множини $\mathfrak{M}$, то виявляється, що обидва перетворення *(AB) C* і *A (BC)* переводять кожен елемент множини $\mathfrak{M}$ в однаковий елемент, тобто обидва перетворення тотожні: *(AB) C = A (BC).*

При дотриманні асоціативного закону добуток великого числа елементів не залежить від послідовності, в якій їх перемножують. Наприклад *((AB)C)D=A(B(CD)).* Тому при записі добутку кількох множників послідовність дій не відзначається. Наприклад, замість *((AB) C)D* можна писати просто *ABCD.*

У якості одиниці (як правої, так і лівої) в групі перетворень служить так зване тотожне перетворення, що залишає кожен елемент множини $\mathfrak{M}$ на місці.

Перетворення, протилежне даному, отримується таким чином: якщо дане перетворення переводить M в M', то зворотним називається перетворення, що переводить M' в *M*. Наприклад, якщо $A = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_1 \end{pmatrix}$, то перетворення, обернене до *A*, переводить x_2 в x_1, x_3 в x_2, x_1 в x_3. Таким чином воно є $\begin{pmatrix} x_2 & x_3 & x_1 \\ x_1 & x_2 & x_3 \end{pmatrix}$ або $\begin{pmatrix} x_1 & x_2 & x_3 \\ x_3 & x_1 & x_2 \end{pmatrix}$.

В теорії груп справедливі такі теореми.

Теорема. Група містить не більше однієї правої одиниці.

Доведення. Нехай J і J_1 будуть дві праві одиниці групи, причому J буде та одиниця, яка потрібна для аксіоми IV. Маємо: $J_1 J = J_1$. Помножимо обидві частини праворуч на J_1 : $J_1 J J_1 = J_1 J_1 = J_1$. Помножимо обидві частини праворуч на елемент *X*, правий зворотний до тобто такий, що $J_1 X = J : J_1 J J_1 X = J_1 X$, тобто $J_1 J = J$. Порівнюючи знайдені вирази, отримуємо: $J_1 = J$ звідки випливає, що обидві одиниці не можуть бути різні.

Теорема. Права одиниця одночасно є і лівою одиницею.

Доведення. Потрібно показати, що для будь-якого елемента *A* групи має місце $JA = A$. Введемо позначення $JA = B$. Якщо помножити цю рівність праворуч на елемент *X*, правий зворотний до *A* $(AX = J) : JAX == BX$, тобто $J = BX$. Таким чином $AX = B$. Множачи праву частину на елемент *Y*, правий зворотний до *X (XY = J)*, можна отримати: *AXY = BXY*, звідки *A = B*, тобто *J A = A.* Це доведення дає право називати елемент J просто одиницею.

Теорема. Правий обернений елемент є водночас і лівим оберненим елементом, тобто з $AX = J$ слідує $XA = J$.

Доведення. Помноживши рівність $XA = B$ праворуч на X: можна отримати: $AX = BX$, тобто $X = BX$. Помноживши праворуч її на елемент Y, правий зворотний до X ($XY = J$), можна отримати: $XY = BXY$, тобто $J = B$, звідки $XA = J$, що й потрібно було довести.

Теорема. Не може існувати двох різних правих обернених елементів.

Доведення. Нехай має місце $AX = AY = J$. Помноживши рівність зліва на X: $XAX = XAY$, можна отримати: $JX = JY$. Звідси $X = Y$, що й потрібно було довести.

3.2.2. Скінченні групи.

Групи, які містять скінчене число елементів, називаються скінченними, а число елементів, що містяться в такій групі, – порядком групи.

Крім порядку групи, існує поняття порядку елемента групи. Перемножуючи елемент A необмежену кількість разів сам на себе, тобто підносячи A до степеня A, A^2, A^3…, в силу скінченності групи, не можна постійно отримувати різні елементи, тобто вони повинні повторюватися. Це означає, що матимуть місце рівності типу $A^{m+k}=A^m$, де $k > 0$. Помноживши праву частину на елемент, обернений до A^m, отримається рівність: $A^k = J$. Найменше з чисел k, які дають $A^k = J$, називається порядком елемента A.

3.2.3. Представлення підстановок у вигляді циклів.

Розглянемо яку-небудь підстановку, наприклад, $\begin{pmatrix}123456789\\314876925\end{pmatrix}$. У ній цифра 1 переходить у 3, 3 – в 4, 4 – у 8, 8 – в 2, 2 – в 1, тобто "цикл замкнувся". Ця кругова заміна цифр називається циклом і позначається так: (1 3 4 8 2). Використовуючи не задіяні попередніми циклами елементи, можна отримаємо ще цикли: (5 7 9) і (6). Таким чином, підстановку в циклах можна записувати так: (1 3 4 8 2), (5 7 9), (6). У цьому випадку говорять, що підстановка складається з 5-членного, 3-членного і 1-членного циклів. Останні, котрі залишають цифру на місці, зазвичай не пишуться.

Від циклічної форми підстановки неважко перейти до початкової. Нехай дано, наприклад, підстановка (1 3) (2 4) (5 6 7). З визначення циклу слідує, що 1 переходить в 3, а 3 – в 1. Тому (1 3) (2 4) (5 6 7) $=\begin{pmatrix}1234567\\3412675\end{pmatrix}$.

Можна розглядати окремі цикли як самостійні підстановки, і тоді неважко побачити, що кожна підстановка є відтворенням складових її циклів.

Розглянемо m-членний цикл (1, 2, 3, ..., m), який являє підстановку, що збільшує кожну цифру на одиницю, коли кратності числа m відкидаються: $x \to x + 1 \pmod m$. Квадратом цієї підстановки є підстановка $x \to x + 2 \pmod m$. Остання підстановка складається при непарному m з одного циклу, а при парному m – з двох циклів. Щоб скласти цей цикл (або цикли), треба в ряді 1,2 ,..., m брати цифри через одну. Таким чином $(1\ 2\ 3\ 4\ 5)^2 = (1\ 3\ 5\ 2\ 4)$, $(1\ 2\ 3\ 4\ 5\ 6)^2 = (1\ 3\ 5)\ (2\ 4\ 6)$. Куб циклу (1, 2, .. m) є підстановка $x \to x + 3 \pmod m$, тобто в циклах цієї підстановки треба брати цифри з ряду 1, 2, .. m через дві т . д.

Оскільки n-ий степінь циклу (1 2 3 ... m) переводить x в $x + n$, причому кратності числа m відкидаються, то показник найменшого степеня m-членного циклу, що дає тотожну підстановку, дорівнює m.

3.3. Підгрупи.

Якщо деяка частина елементів групи утворює самостійну групу, то вона носить назву підгрупи або дільника первісної групи.

Приклад. Симетрична група підстановок 3-го степеня (тобто з 3 предметів) складається з наступних підстановок: 1, (1 2 3), (1 3 2), (1 2), (1 3), (2 3).

Її можна отримати, якщо написати різноманітні підстановки, в яких у верхньому ряді буде стояти послідовність 1 2 3, а в нижньому – всі можливі перестановки з цифр 1, 2, 3. Порядок групи буде дорівнювати 6. Її підгрупи:

a) 1, (1 2 3), (1 8 2) – порядку 3;
b) 1, (1 2) – „ 2;
c) 1, (1 3) – „ 2;
d) 1, (2 3) – „ 2;
e) 1 – „ 1 (тотожна група).

Сукупності декількох елементів групи позначаються великими готичними літерами: $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$... Якщо сукупність $\mathfrak{A}$ складається з елементів $A_1, A_2, \ldots, A_m$, то прийнято позначати сукупність, як суму своїх елементів: $\mathfrak{A} = A_1 + A_2 + \ldots + A_m$. Точно так само, поєднуючи кілька сукупностей, їх можна з'єднувати знаком +. Якщо в отриману сукупність який-небудь елемент увійде кілька разів, то писати його потрібно тільки один раз. Наприклад, якщо $\mathfrak{A} = A + B$, $\mathfrak{B} = B + C$, то $\mathfrak{A} + \mathfrak{B} = A + B + C$.

Під добутком двох сукупностей будемо розуміти наступне. Якщо $\mathfrak{A} = A_1 + A_2 + \ldots + A_m$, $\mathfrak{B} = B_1 + B_2 + \ldots + B_m$, то добуток $\mathfrak{AB}$ є сукупність елементів A_iB_k ($i = 1, 2, \ldots, m$; $k = 1, 2, \ldots n$). Якщо при цьому серед елементів A_iB_k попадуться однакові, то їх слід брати по одному разу, тобто зайві відкидати.

Зокрема, якщо $n = 1$, тобто $\mathfrak{B} = B_1$, то $\mathfrak{A}\, B_1 = (A_1 + A_2 + \ldots + A_m)B_1 = A_1 B_1 + A_2 B_1 + \ldots + A_m B_1$.

Справедлива така

Теорема. Сукупність $\mathfrak{A}$ складає групу тоді і тільки тоді, коли має місце рівність $\mathfrak{A}\mathfrak{A} = \mathfrak{A}$.

Доведення. Якщо $\mathfrak{A}$ складає групу, то добуток $\mathfrak{A} \cdot \mathfrak{A}$ не може містити нічого іншого, крім елементів з $\mathfrak{A}$, з іншого боку, воно неодмінно містить всі елементи з $\mathfrak{A}$, так, як група $\mathfrak{A}$ містить одиницю J, а тому $\mathfrak{A} \cdot \mathfrak{A}$ містить $\mathfrak{A} \cdot J$, тобто $\mathfrak{A}$. Навпаки, якщо $\mathfrak{A}\mathfrak{A} = \mathfrak{A}$, то $\mathfrak{A}$ теж є групою.

Нехай $\mathfrak{G} = A_1 + A_2 + \ldots + A_n$ є група, $\mathfrak{H}$ – її підгрупа. Сукупності типу $\mathfrak{H}A_i$ називаються спряженими системами або суміжними класами. Має місце

Теорема. Системи $\mathfrak{H}A_i$ містять при всякому A_i одну і ту ж саму кількість елементів (рівну порядку групи $\mathfrak{H}$). Дві системи $\mathfrak{H}A_i$ і $\mathfrak{H}A_k$ або збігаються, або не містять загальних елементів.

Доведення. 1. Щоб довести, що числа елементів сукупностей $\mathfrak{H}$ і $\mathfrak{H}A_i$ однакові, достатньо показати, що серед елементів $\mathfrak{H}A_i = A_i + B_2A_i + \ldots + B_mA_i$, де $\mathfrak{H} = 1 + B_2 + \ldots + B_m$, не зустрічається однакових. Припускаючи наприклад, що має місце $B_rA_i = B_sA_i$ при $r \neq s$, і домножуючи це рівняння праворуч на A_i^{-1}, можна отримати $B_r = B_s$, що неможливо, так як всі елементи $1, B_2, \ldots, B_m$ різні.

Нехай $\mathfrak{H}A_i$ і $\mathfrak{H}A_k$ мають загальний елемент $B_rA_i = B_sA_k$. Тоді, множачи її зліва на B_s^{-1}, можна отримати: $A_k = B_s^{-1} B_r \cdot A_i$.

Якщо взяти довільний елемент B_tA_k системи $\mathfrak{H}A_k$, то він дорівнює $B_tB_s^{-1}$. Але так як $B_tB_s^{-1}B_r$ є елемент групи $\mathfrak{H}$, то B_tA_k є елемент системи $\mathfrak{H}A_i$ Подібним же чином можна показати, що будь-який елемент системи $\mathfrak{H}A_i$ міститься в $\mathfrak{H}A_k$. Таким чином $\mathfrak{H}A_i = \mathfrak{H}A_k$.

Теорема (Лагранжа). Порядок підгрупи є дільником порядку первісної групи.

Доведення. Нехай, як і раніше

$\mathfrak{G} = 1 + A_2 + \ldots + A_n$, $\mathfrak{H} = 1 + B_2 + \ldots + B_m$.

Тут порядок $\mathfrak{G}$ є n і порядок $\mathfrak{H}$ є m. В силу того, що елементи B_i містяться серед A_j, а також того, що $\mathfrak{H}$ містить одиницю, будемо мати:

$\mathfrak{H} + \mathfrak{H}A_2 + \ldots + \mathfrak{H}A_n = \mathfrak{H}\mathfrak{G} = \mathfrak{G}$.

У лівій частині цієї рівності залишаються тільки по одній з однакових спряжених систем, а решту викреслюються. Нехай у результаті вийде розкладання $\mathfrak{G} = \mathfrak{H} + \mathfrak{H}A_2 + \ldots + \mathfrak{H}A_k$, причому всі елементи правої частини різні між собою. Тому в правій частині рівності міститься mk різних елементів, а в лівій їх всього n. Тому $n = mk$.

Ця рівність доводить теорему.

Наслідок. Порядок групи поділяється на порядок будь-якого свого елемента.

Групи, утворені степенями якого-небудь елемента, називаються циклічними. Всі циклічні групи абелеві, тобто підкоряються комутативному закону.

3.3.1. Розклад підстановок на транспозиції.

Будь-яку підстановку можна представити у вигляді добутку декількох транспозицій, тобто підстановок, що переміщують дві цифри і залишають інші цифри незмінними. Оскільки будь-яку підстановку можна представити у вигляді добутку циклів, то досить перевірити це твердження для циклу, що є неважко. Насправді,

(1 2 3) = (1 2) (1 3),

(1 2 3 4) = (1 2) (1 3) (1 4),

……

(12 3 ... m) = (1 2) (1 3) ... (1 m).

Потрібно звернути увагу на те, що цикли з парною кількістю цифр розкладаються на непарне число транспозицій, з непарним числом цифр – на парне число транспозицій. Тому для кожної даної підстановки можна вирахувати, чи розкладеться вона на парне, або на непарне число транспозицій. Дійсно, якщо підстановка складається з циклів порядків n_1, n_2,…, n_k ($n_1 + n_2 + ... + n_k = n$), то парність отриманого числа транспозиції залежить від того, чи буде число $(n_1 - 1) + (n_2 - 1) + \ldots + (n_k - 1) = n - k$ парним чи непарним. При цьому треба також брати до уваги і одночленні цикли.

Парність підстановки не залежить від способу її розкладання на транспозиції. Кожну підстановку можна розкласти на транспозиції не одним, а багатьма способами. При цьому парність числа отриманих транспозицій залишається однією і тією ж. Найпростіше переконатися в цьому можна, якщо звернути увагу, що кожна транспозиція змінює знак виразу:

$$(x_1 - x_2)\cdot(x_1 - x_3)\cdot(x_2 - x_3)\cdot\ldots\cdot(x_{n-1} - x_n) = \prod_{i,k}(x_i - x_k). \qquad (3.1)$$

Потрібно розглянути, як на вираз (3.1) діє транспозиція $(x_1 - x_3)$. Для цього вираз (3.1) трансформується:

$$(x_1 - x_2)\prod_{i=3}^{n}(x_1 - x_3)\prod_{i=3}^{n}(x_2 - x_3)\prod_{i,k}(x_i - x_k), \qquad (3.2)$$

де в останньому добутку значки i та k пробігають значення 3, 4, .. n. Транспозиція (x_1, x_2) змінює знак першого множника $(x_1 - x_2)$ і перетворює друге множення у третє, третє в друге, а на останнє ніяк не впливає. Тому

весь добуток (3.2) змінить знак. Це твердження доводиться аналогічно для будь-якої іншої транспозиції.

Якщо в такий спосіб підстановка розкладається на парне число транспозиції, то вона не змінить знака у виразі (3.1); якщо ж на непарне, то змінить. Тому підстановка цілком визначає собою парність числа транспозицій, на які вона розкладається. Підстановку прийнято називати парною або непарною, в залежності від того, залишає вона вираз (3.1) незмінним чи змінює його знак.

Неважко довести

Теорему. Відтворення двох парних або двох непарних підстановок дає парну підстановку; відтворення ж парної підстановки на непарну дає непарну підстановку.

3.3.2. Симетричні групи.

З n цифр 1, 2, ..., n можна утворити $n! = 1.2 \ldots n$ різних перестановок. Нехай $a_1, a_2, a_3, \ldots a_n$ буде однією з них. Можна побудувати відповідну їй підстановку $\begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ a_1 & a_2 & a_3 & \ldots & a_n \end{pmatrix}$. Різним перестановкам будуть відповідати різні підстановки. Таким чином можна отримати всього $n!$ різних підстановок, які, очевидно, утворюють групу, яка називається симетричною з n цифр або симетрично групою n-го степеня і позначається так: $\mathfrak{S}_n$.

Розглянемо довільну групу підстановок n-го степеня (тобто з n цифр). Вона є дільником симетричної групи n-го степеня, а тому її порядком є дільник числа $n!$ Таким чином отримується

Теорема. Порядком групи підстановок n-го степеня є дільник числа $n!$.

3.3.3. Знакозмінні групи.

Всі парні підстановки n-го степеня складають, групу, яка носить назву знакозмінної n-го степеня і позначається так: $\mathfrak{A}_n$. Щоб визначити її порядок потрібно взяти довільну непарну підстановку (наприклад, транспозицію) T. Слід зазначити, що всі підстановки спряженої системи $\mathfrak{A}_n T$ непарні. З іншого боку, будь-яка непарна підстановка S входить в систему $\mathfrak{A}_n T$, оскільки підстановка ST^{-1},в силу чинності теореми 8, парна, і тому входить до групи $\mathfrak{A}_n$. Таким чином можна отримати: $\mathfrak{S}_n = \mathfrak{A}_n + \mathfrak{A}_n T$.

Теорема. Будь-яку підстановку знакозмінної групи можна представити як добуток тричленних циклів.

Доведення. Так як кожну підстановку знакозмінної групи можна представити, як відтворення парного числа транспозиції, то достатньо довести, що відтворення двох транспозицій можна представити, як відтворення тричленних циклів. Тут може зустрітися один з двох випадків: або обидві транспозиції містять загальну цифру, або всі вхідні в них цифри різні.

У першому випадку: (1 2) (2 3) = (1 3 2).

У другому випадку: (1 2) (3 4) = (1 4 3) (1 4 2).

Приклад. Симетрична група $\mathfrak{S}_4$ 4-го степеня складається з наступних підстановок, кількість яких 4! = 24:

$$\begin{cases} 1,(1\ \ 2),(1\ \ 3),(2,\ \ 3),(2\ \ 4),(3\ \ 4),(1\ \ 2),(3\ \ 4),(1\ \ 3)\ \ (2\ \ 4),(1\ \ 4)(2,\ \ 3), \\ (1\ \ 2\ \ 3),(1\ \ 3\ \ 2),(1\ \ 2\ \ 4),(1\ \ 4\ \ 2),(1\ \ 3\ \ 4),(1\ \ 4\ \ 3),(2\ \ 3\ \ 4),(2\ \ 4\ \ 3), \\ (1\ \ 2\ \ 3\ \ 4),(1\ \ 2\ \ 4\ \ 3),(1\ \ 3\ \ 2\ \ 4),(1\ \ 4\ \ 2\ \ 3),(1\ \ 4\ \ 3\ \ 2) \end{cases}$$

Знакозмінна група $\mathfrak{A}_4$ складається з 12 підстановок:

$$\begin{cases} 1,(1\ \ 2)(3\ \ 4),(1\ \ 3)(2\ \ 4),(1\ \ 4)(2\ \ 3),(1\ \ 2\ \ 3),(1\ \ 3\ \ 2),(1\ \ 2\ \ 4),(1\ \ 4\ \ 2), \\ (1\ \ 3\ \ 4),(1\ \ 4\ \ 3),(2\ \ 3\ \ 4),(2\ \ 4\ \ 3). \end{cases}$$

Крім того, група $\mathfrak{S}_4$ містить наступну підгрупу порядку 8:

1, (1 3 2 4), (1 4 2 3), (1 2) (3 4), (1 3) (2 4), (1 4) (2 3), (1 2), (3 4).

Цю групу можна охарактеризувати, як сукупність підстановок, що не змінюють виразу $x_1x_2+x_3x_4$.

Ця група містить наступну (абелеву) підгрупу порядку 4:

1, (1 2) (3 4), (1 3) (2 4), (1 4) (2 3).

Крім того, $\mathfrak{S}_4$ містить кілька циклічних груп, а також групу:

1, (1 2), (3 4), (1 2) (3 4).

Тут вперше зустрічаються приклади абелевих, але не циклічних груп.

3.3.4. Транзитивні групи підстановок.

Група підстановок називається транзитивною, якщо в ній містяться підстановки, що переводять будь-які дві цифри одна в одну. В іншому випадку група підстановок називається інтранзитивною.

Нехай $\mathfrak{G}$ – інтранзитивна група підстановок n-го степеня і її підстановки переводять цифру 1 в кожну з цифр 1, 2, …, m. Тоді $\mathfrak{G}$ міститиме підстановки, що переводять одну в одну будь-які дві цифри з ряду 1, 2, …,m. Насправді, якщо S_1 переводить 1 у r, а S_2 – 1 в s, то підстановка $S_1^{-1}S_2$ переведе r в s. Таким чином на підставі інтранзитивності групи $\mathfrak{G}$ ряд 1, 2, …, m не вичерпує всіх цифр 1, 2, ... , n. Беручи з ряду 1, 2, ..., n яку-небудь, цифру, що не належить до ряду 1, 2, ..., m, можна переконатися, що підстановки групи $\mathfrak{G}$ будуть переводити її в цифри, що не входять в ряд 1, 2,

..., m. Ряду таких цифр притаманні ті ж властивості, що і ряду 1, 2, ..., m. Продовжуючи виділення таких систем цифр, можна розбити весь ряд 1, 2, ..., n на окремі системи, що носять назву систем інтранзитивності групи $\mathfrak{G}$. Ці системи характеризуються тим, що: 1) група $\mathfrak{G}$ містить підстановки, що переводять одну в одну цифри однієї і тієї ж системи інтранзитивності, 2) група $\mathfrak{G}$ не містить підстановок, що переводять одну в одну цифри різних систем.

Група називається k разів транзитивною, якщо в ній міститься підстановка, що переводить будь-які k з цифр 1, 2, ... , n в будь-які задані цифри того ж ряду. Симетрична група n-го степеня n разів транзитивна тому, що містить підстановку, яка переводить кожну з цифр 1, 2, ..., n в цю ж сукупність цифр, взяту в будь-якому заданому розташуванні.

Можна показати, що знакозмінна група n-го степеня n – 2 рази транзитивна. Дійсно, нехай потрібно перевести цифри 1, 2,. .., n – 2 в цифри $a_1, a_2, ..., a_{n-1}$, довільно вибрані з ряду 1, 2, ..., n. Нехай $a_1, a_2, ..., a_{n-1}, a_n$ буде деяка перестановка цифр 1, 2, ..., n. Тоді обидві підстановки:

$$\begin{pmatrix} 1 & 2 & 3, ..., n-2, n-1, n \\ a_1, & a_2, & a_3, ..., a_{n-2}, a_{n-1}, a_n \end{pmatrix} i \begin{pmatrix} 1 & 2 & 3, ..., n-2, n-1, n \\ a_1, & a_2, & a_3, ..., a_{n-2}, a_{n-1}, a_n \end{pmatrix} (a_{n-1}, a_n) =$$

$$= \begin{pmatrix} 1 & 2 & 3, ..., n-2, n-1, n \\ a_1, & a_2, & a_3, ..., a_{n-2}, a_{n-1}, a_n \end{pmatrix}$$

переводять *1, 2 ,..., n–2* у $a_1, a_2, ..., a_{n-2}$. У той же час. одна з цих підстановок в силу теореми 8, парна і тому міститься в знакозмінній групі.

Розглянемо групу $\mathfrak{G}$ підстановок n-го степеня, і хай її система інтранзитивності, що містить цифру 1, складається з m цифр 1, 2, ..., m. Виділимо в ній сукупність $\mathfrak{H}$ підстановок, що залишають одну з цифр, наприклад 1, на місці. Неважко переконатися, що сукупність, $\mathfrak{H}$ становить групу. У групі $\mathfrak{G}$ в силу визначення системи інтранзитивності знайдуться підстановки $S_1 = 1, S_2, S_3 ..., S_m$, що переводять 1 відповідно в 1, 2, 3 ,..., m. Тоді всі підстановки спряженої системи $\mathfrak{H}$ S_i (i = 1, 2 ,..., m) переводять 1 в i. У силу цього ці спряжені системи не містять загальних підстановок.

Теорема. Підстановки транзитивної абелевої групи розкладаються на цикли рівних порядків.

Доведення. Нехай яка-небудь транзитивна Абелева група $\mathfrak{H}$ містить підстановку S_1 , що складається з циклів різних порядків. Якщо f є порядком найменшого з її циклів, то $S=S_1$ міститиме і цифри, що залишаються на місці, і справжні цикли. Треба показати, що це неможливо. Нехай S переводить 1 в 2 і залишає 3 на місці. У силу транзитивності у групі $\mathfrak{H}$ знайдеться підстановка T, переводячи 3 в 1. Умова $ST=TS$ переставних підстановок S і T дає

$$\begin{pmatrix}1 & 3 & \dots\\ 2 & 3 & \dots\end{pmatrix}\begin{pmatrix}3 & \dots\\ 1 & \dots\end{pmatrix}=\begin{pmatrix}3 & \dots\\ 1 & \dots\end{pmatrix}\begin{pmatrix}1 & 3 & \dots\\ 2 & 3 & \dots\end{pmatrix}.$$

Але ця рівність неможлива, так як її ліва частина переводить 3 в 1, в той час, як її права частина переводить 3 в 2.

Теорема. Порядок транзитивної абелевої групи $\mathfrak{G}$ підстановок дорівнює її степеню, тобто числу переміщуваних нею цифр.

Доведення. Порядок групи $\mathfrak{G}$ дорівнює її степеню n, помноженому на порядок її найбільшої підгрупи $\mathfrak{H}$, що залишає одну з цифр на місці. Але так як $\mathfrak{G}$ – абелева група, то підстановки групи $\mathfrak{H}$ повинні залишати всі цифри на місці, тобто $\mathfrak{H}$ складається тільки з тотожної підстановки, і її порядок дорівнює одиниці. Звідси випливає, що порядок групи $\mathfrak{G}$ дорівнює n.

3.4. Імпримітивні групи.

Група підстановок називається імпримітивною, якщо всі переміщувані нею цифри можна розбити на такі системи (названі системами імпримітивності), що кожна підстановка групи переводить цифри кожної системи в цифри якої-небудь певної системи. Іншими словами, якщо дві цифри належать одній і тій же системі, то кожна підстановка групи переводить їх у цифри, що належать теж одній і тій же системі.

На практиці зручно перевіряти імпримітивність так: позначаємо всі цифри однієї (кожної) системи імпримітивності одним і тим же знаком. Якщо одержані таким чином підстановки не містять протиріч, то позначені одним знаком цифри дійсно складають систему імпримітивності групи; в іншому випадку – не складають.

Приклад. Група 8–го порядку

1, (1 2) (3 4), (1 3) (2 4), (1 4) (2 3), (1 2), (3 4), (1 3 2 4), (1 4 2 3)

має системи 1, 2 і 3, 4 в якості систем імпримітивності. Щоб переконатися в цьому, треба позначити цифри 1, 2 знаком I, а цифри 3, 4 – знаком II. Тоді отримаємо наступні заміни:

1, (I I) (II II), (I II) (I II), (I II) (I II), (II) (II) (II), (I) (I) (II П), (I II I II), (I II I II).

Всі ці підстановки зводяться до двох наступних: 1, (I II), і суперечності ніде немає.

Системи ж 1, 3 і 2, 4 не можуть слугувати системами імпримітивності групи. Дійсно, позначаючи 1, 3 через a, а 2, 4 через b, можна отримати такі підстановки:

1, *(ab) (ab), (aa) (bb), (ab) (ba), (ab) (a) (b), (a) (b) (ab), (aabb), (abba)* .

Перші чотири з цих підстановок до протиріччя не приводять, а тому для утвореної ними групи системи 1, 3 і (2, 4) є системами імпримітивності. Останні ж чотири з цих підстановок містять суперечності. Наприклад,

підстановку (*a a b b*) можна переписати так: $\begin{pmatrix} a & a & b & b \\ a & b & b & a \end{pmatrix}$ З одного боку, *a* переходить в *a*, з іншого - *a* переходить у *b*.

Теорема. Всяка транзитивна група, що містить транспозицію, є симетричною, або імпримітивною групою.

Доведення. Нехай група містить транспозицію (12). Якщо вона містить усі транспозиції, що переводить 1 в кожну з переміщуваних цифр 2, 3 ,..., n, то вона є симетричною групою з *n* цифр. Дійсно, *(rs) = (1r) (1s) (1r)*, і група, містячи кожну з транспозицій правої частини рівності, повинна містити і транспозицію *(rs)*. Але будь-яку підстановку можна представити у вигляді добутку транспозицій. Таким чином, група містить усі підстановки симетричної групи.

Припустимо тепер, що група містить не всі підстановки типу *(1r)*, а тільки деякі з них, наприклад *(1 2), (13),..., (1m)*. Доведемо, що система *1, 2 ,..., m* є для нашої групи системою імпрімітивності. У силу транзитивності групи в ній існують підстановки, що переводять кожну з цифр *1, 2 ,..., m* в будь-яку з цифр ряду *m + 1 ,..., n*. Нехай T буде підстановка, що переводить приклад 1 в цифру *λ*, котра не належить до ряду *1, 2 ,..., m*. Тоді жодна з цифр ряду *1, 2,..., m* не може перейти у цифру цього ж ряду. Припустимо протилежне, наприклад, що 2 переходить в 3. Тоді в групі буде міститися підстановка

$$T^{-1}\,(12)\,T = \begin{pmatrix} \lambda & 3 & \dots \\ 1 & 2 & \dots \end{pmatrix}\ (1\ 2)\ \begin{pmatrix} \lambda & 2 & \dots \\ 1 & 3 & \dots \end{pmatrix} = (\lambda\ 3),$$

а також підстановка *(1 3)(λ 3)(l 3)=(1 λ)*, що суперечить припущенню.

Нехай якась підстановка T переводить *λ1, λ 2,..., λm*. в *λ1, λ 2,..., λm*. Розмірковуючи подібно попередньому, можна переконатися, що дана група містить всі транспозиції *(λi, λk)* і не містить транспозиції, що переводять наприклад λ1 в цифри, які не належать до системи *λ1, λ 2,..., λm*. Тому можна взяти цю систему в якості вихідної системи *1, 2 ,..., m*. Так можна переконатися, що цифри *λ1, λ 2,..., λm* або не виводяться із системи, або переводяться в цифри, які не належать до системи *λ1, λ 2,..., λm*. Але система *λ1, λ 2,..., λm* перекладається різними підстановка нашої групи в ті самі системи, що й система *2 ,..., m*. Дійсно, якщо підстановка S переводить *λ1, λ 2,..., λm* в *μ1, μ 2,..., μm,* то підстановка *TS* переводить *1, 2 ,..., m в μ1, μ 2,..., μm*. У силу цього жодна із систем, що отримується з *1, 2 ,..., m* за допомогою різних підстановок групи, не має загальних цифр з іншими подібними системами. Разом з тим, у силу транзитивності групи усі ці системи вичерпують всі цифри *1,2, ..., n*. Це доводить імпрімітівність групи.

3.4.1. Нормальні дільники.

Якщо елемент A деякої групи помножити праворуч на елемент B тієї ж групи, а ліворуч - на B^{-1}, то отриманий елемент $B^{-1}AB$ називається елементом, перетвореним з A за допомогою B або спряженим з елементом A.

Перетворення не змінює елемента тоді і тільки тоді, якщо перетворюючий і перетворюваний елементи переставні. Дійсно, з рівності $B^{-1}AB$ випливає $BA = AB$ і навпаки.

Теорема. Порядки спряжених елементів рівні.

Доведення. Якщо $\bar{A} = B^{-1}AB$, то $\bar{A}^2 = B^{-1}AB.\ B^{-1}AB = B^{-1}A^2\ B$; далі, $\bar{A}^3 = \bar{A}^2.\bar{A} = B^{-1}A^2B.\ B^{-1}AB = B^{-1}A^3B$, і т. д. Якщо $A^m = J$, то $\bar{A}^m = B^{-1}A^mB = B^{-1}JB = J$. Тому порядок $\bar{A}$ не може бути вище порядку A. Зворотно, якщо $\bar{A} = B^{-1}AB$, то звідси $A = B\bar{A}B^{-1}$, і таким чином, якщо $\bar{A}^m = J$ той $\bar{A}^m = B\bar{A}^mB^{-1} = BJB^{-1} = J$. Отже, порядки обох елементів рівні.

Якщо елементи групи є підстановками, і притому підстановка A записана в циклах, то неважко, не виконуючи дій, написати перетворену підстановку. Для цього лише потрібно зробити підстановку B над цифрами, що виражають A в циклах. Справді, нехай

$A = (1,2,3,...,k)(k+1,...,l)...(m+1,.:.,n)$, $B = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ \alpha 1 & \alpha 3 & \alpha 2 & \dots & \alpha n \end{pmatrix}$,

тоді

$$B^{-1}AB = \begin{pmatrix} a1 & a3 & \dots & an \\ 1 & 2 & a2 & \dots & n \end{pmatrix} (1\ 2\ 3\ \dots,\ k)(k+1,\dots,l)\dots(m+1,.:.,n).$$

$$\begin{pmatrix} 1 & 2 & \dots & n \\ a1 & a3 & \dots & an \end{pmatrix}.$$

Нехай в A поруч з цифрою i стоїть k. Тоді B^{-1} переводить α_i в i, A переводить i в k, B переводить k в α_k. Таким чином $B^{-1}AB$ переводить α_i в α_k, і можна отримати:

$$B^{-1}AB = (\alpha 1\alpha 2\alpha 3\dots\alpha k)(\alpha k+1,\dots,\alpha l)\dots(\alpha m+1,\dots,\alpha n).$$

Це ще раз підтверджує той факт, що порядок підстановки не змінюється від перетворення: дійсно, порядок підстановки є найменше кратне від порядків її циклів, у той час, як від перетворення цикли залишаються незмінними.

3.4.2. Перетворення сукупностей і груп.

Під перетворенням сукупності елементів групи ми будемо розуміти одночасне перетворення всіх елементів сукупності за допомогою одного і

того самого елемента. Якщо $\mathfrak{M}$ - задана сукупність, то перетворену за допомогою елемента B сукупність можна позначати так: $B^{-1}\,\mathfrak{M}B$. Нехай $\mathfrak{M} = A_1 + A_2 + \ldots + A_m$; тоді $B^{-1}\mathfrak{M}B = B^{-1}A_1B + B^{-1}A_2B + \ldots + B^{-1}A_mB$.

Теорема. Якщо сукупність $\mathfrak{M}$ утворює групу (підгрупу вихідної групи), то й перетворена сукупність $B^{-1}\mathfrak{M}B$ утворює групу.

Доведення. Нехай A_i, A_j, A_τ будуть елементами групи $\mathfrak{M}$, пов'язані співвідношенням $A_iA_j = A_\tau$ Тоді $(B^{-1}A_iB)\ (B^{-1}A_jB) = B^{-1}A_\tau\ B$. Таким чином, добуток елементів, що входять до $B^{-1}\mathfrak{M}B$, теж входить до $B^{-1}\mathfrak{M}B$. Далі, одиниця перетворюється в одиницю: $B^{-1}JB = J$. Нарешті, зворотний елемент перетвориться у зворотний: $(B^{-1}AB) = B^{-1}A^{-1}B$.

Дійсно, $B^{-1}\,AB\,B^{-1}A^{-1}\,B = B^{-1}\,A^{-1}B = B^{-1}\,B = J$. Таким чином сукупність $B^{-1}\mathfrak{M}B$ утворює групу.

Групи $\mathfrak{M}$ і $B^{-1}\mathfrak{M}B$ мають однакові порядки, так як два різних елемента групи $\mathfrak{M}$ перетворюються неодмінно в різні елементи: якби, наприклад, виконувалася б рівність $B^{-1}\,AB = B^{-1}\,CB$, то, множачи зліва на B, а справа на B^{-1}, отримали б $A = C$.

Групи, одержані одна з одної за допомогою перетворення носять назву спряжених. Якщо група $\mathfrak{H}$ є дільником групи $\mathfrak{G}$ і якщо при цьому всі спряжені з $\mathfrak{H}$ підгрупи, одержані з $\mathfrak{H}$ перетвореннями за допомогою всіляких елементів групи $\mathfrak{G}$ збігаються, то $\mathfrak{H}$ називається нормальним дільником (або інваріантною підгрупою) групи $\mathfrak{G}$.

Приклад. Розглянемо симетричну групу $\mathfrak{S}_4$ 4-го степеня. Її підгрупа 1, (1 2) (3 4), (1 3) (2 4), (1 4) (2 3), є нормальним дільником групи $\mathfrak{S}_4$. Це відразу випливає з того, що група $\mathfrak{S}_4$ містить тільки 3 подвійних транспозиції. Тому будь-яке перетворення, зберігаючи цикловий тип підстановки, неодмінно переводить сукупність цих підстановок в себе. Точно так само ця група є нормальним дільником інших підгруп групи $\mathfrak{S}_4$, а саме знакозмінної групи $\mathfrak{A}_4$ і наступної підгрупи 8-го порядку:

1, (1 2) (3 4), (1 3) (2 4), (1 4) (2 3), (1 2), (3 4), (1 3 2 4), (1 4 2 3) (3.8)

Ця група вже не є нормальним дільником симетричної групи. Щоб переконатися в цьому, досить перетворити її підстановку (1 2) за допомогою підстановки (2 3) групи $\mathfrak{S}_4$: (2 3) (1 2) (2 3) = (1 3).

На цих прикладах можна переконатися, чго перетворення залишають незмінними і всю сукупність елементів в нормальному дільнику, але кожен окремий елемент може і не залишатися незмінним при перетвореннях.

Теорема. Знакозмінна група є нормальним дільником симетричної групи.

Доведення. Якщо підстановка A входить до знакозмінної групи, то вона є парною підстановкою. Нехай B – довільна підстановка симетричної групи. Тоді підстановка $B^{-1}AB$ парна, тобто входить до знакозмінної групу.

Теорема. Будь-який дільник абелевої групи є нормальним дільником.

Доведення. Це видно з того, що кожен елемент абелевої групи не змінюється від перетворення за допомогою іншого елемента абелевої групи.

Нехай $\mathfrak{G}$ – скінченна група, $\mathfrak{H}$ – її підгрупа. Виникає питання, скільки існує підгруп, спряжених з $\mathfrak{H}$?

Щоб відповісти на нього, розглянемо сукупність $\mathfrak{N}$ таких елементів N групи $\mathfrak{G}$, що $N^{-1}\mathfrak{H}N = \mathfrak{H}$. Неважко побачити, що сукупність $\mathfrak{N}$ є група. Дійсно, якщо N_1 і N_2 входять у $\mathfrak{N}$, то $(N_1 N_2)^{-1}\mathfrak{H}(N_1 N_2)= N_2^{-1} (N_1^{-1} \mathfrak{H} N_1) N_2$, звідки слідує, що й $N_1 N_2$ входить у $\mathfrak{N}$. Подібним же чином можна показати, що одиниця входить до $\mathfrak{N}$, і що елемент, обернений до елементу з $\mathfrak{N}$, входить в $\mathfrak{N}$. Група $\mathfrak{N}$ носить назву нормалізатора підгрупи $\mathfrak{H}$. Тому має місце

Теорема. Нехай $\mathfrak{H}$ є дільником групи $\mathfrak{G}$. Число підгруп, спряжених з $\mathfrak{H}$, дорівнює індексу нормалізатора $\mathfrak{N}$ підгрупи $\mathfrak{H}$ відносно $\mathfrak{G}$.

Доведення. Розкладемо групу $\mathfrak{G}$ на спряжені системи по підгрупі $\mathfrak{N}$:

$\mathfrak{G}= \mathfrak{N} + \mathfrak{N}S_2+ \mathfrak{N}S_3+ \ldots +\mathfrak{N}S_k$.

Тоді єдиними групами, спряженими, з $\mathfrak{H}$ будуть наступні:

$\mathfrak{H}_1 = \mathfrak{H}$, $\mathfrak{H}_2= S_2^{-1}\mathfrak{H} S_2$, $\mathfrak{H}_3= S_3^{-1}\mathfrak{H} S_3,\ldots, \mathfrak{H}_k$.

Дійсно, будь-який елемент A групи $\mathfrak{G}$ може бути представлений у формі $\mathfrak{N}S_i$, де N – елемент групи $\mathfrak{N}$. Тому $A^{-1}\mathfrak{H}A=(NS_i)^{-1}\mathfrak{H}(NS_i) =S_i^{-1}\mathfrak{H}(N^{-1}\mathfrak{H} N)S_i$.

Але в силу визначення нормалізатора має місце $N^{-1}\mathfrak{H}N = \mathfrak{H}$, звідки $A^{-1}\mathfrak{H}A = = S_i^{-1}\mathfrak{H}S_i=\mathfrak{H}_i$.

З іншого боку, всі ці групи різні. Справді, припустимо, що має місце

$S_i^{-1}\mathfrak{H} S_i = S_j^{-1}S_j$

Цю рівність можна переписати так:

$(S_iS_j^{-1})^{-1}\mathfrak{H}(S_iS_j^{-1})= \mathfrak{H}$.

Ця рівність вказує на те, що елемент $S_iS_j^{-1}$входить у нормалі затор $\mathfrak{N}$: $S_iS_j^{-1}=N$, звідки $S_i= NS_j$, що суперечить принципу розкладання груп на спряжені системи.

3.4.3. Доповняльні групи.

Нехай $\mathfrak{H}$ є нормальний дільник групи $\mathfrak{G}$ індексу k. Можна побудувати особливу групу порядку k, яка називається доповняльною. Для цього потрібно розкласти $\mathfrak{G}$ на спряжені системи по підгрупі $\mathfrak{H}=\mathfrak{H} + \mathfrak{H}A_2+\ldots+\mathfrak{H}A_k$.

Спряжені системи $\mathfrak{H}A_i$, (i = 1, 2, ... k) можна вважати елементами нової групи, для яких встановлюється спосіб композиції, що випливає з правила композиції сукупностей. Потрібно розглянути добуток $\mathfrak{H}A_i\cdot\mathfrak{H}A_j$ двох довільно взятих спряжених систем. У силу нормальності підгрупи $\mathfrak{H}$

має місце $A_i^{-1}\mathfrak{H}A_i=\mathfrak{H}$, звідки $A_i\mathfrak{H}=\mathfrak{H}A_i$, внаслідок чого відбудеться перетворення: $\mathfrak{H}A_i\cdot\mathfrak{H}A_j=\mathfrak{H}\mathfrak{H}A_iA_j=\mathfrak{H}A_iA_j$. Нехай елемент A_iA_j входить в l-у спряжену систему розкладання $\mathfrak{G}$ по $\mathfrak{H}$ і у зв'язку з цим виражається так: $A_iA_j=HA_l$, де H - елемент групи $\mathfrak{H}$. Тоді можна одержати: $\mathfrak{H}A_i\mathfrak{H}A_j=\mathfrak{H}HA_l=\mathfrak{H}A_l$.

Ця рівність цілком визначає правило множення спряжених систем $\mathfrak{H}\mathrm{A}_i$.Одиничним елементом є підгрупа $\mathfrak{H}$. Зворотний елемент також неважко визначити. Таким чином, спряжені системи утворюють групу порядку k, яку називають доповняльною і позначають так: $\mathfrak{G}/\mathfrak{H}$.

Нехай $\mathfrak{H}_1$ і $\mathfrak{H}_2$ будуть дільниками скінченої групи $\mathfrak{G}$. Сукупність елементів, що входять одночасно в $\mathfrak{H}_1$ і в $\mathfrak{H}_2$ очевидно утворює групу, яку прийнято називати перерізом або найбільшим спільним дільником груп $\mathfrak{H}_1$ і $\mathfrak{H}_2$. Аналогічно визначається переріз більшого числа груп.

Теорема. Якщо $\mathfrak{H}_1$ і $\mathfrak{H}_2$ нормальні дільники групи $\mathfrak{G}$, то і їх переріз $\mathfrak{R}$ є нормальним дільником групи $\mathfrak{G}$.

Доведення. Досить довести, що довільний елемент К групи $\mathfrak{R}$, будучи перетворений за допомогою довільного елемента А групи $\mathfrak{G}$, входить до $\mathfrak{R}$. Згідно з визначенням групи $\mathfrak{R}$, елемент К входить і в $\mathfrak{H}_1$ і в $\mathfrak{H}_2$.В силу нормальності останніх груп елемент A^{-1} К А теж входить в $\mathfrak{H}$ і в $\mathfrak{H}_2$, а тому і в $\mathfrak{R}$.

Неважко поширити цю теорему на перетин більшого числа груп.

Теорема. Нехай $\mathfrak{H}$ – довільний дільник групи $\mathfrak{G}$. Тоді переріз $\mathfrak{R}$ всіх груп, спряжених з $\mathfrak{H}$, є нормальним дільником групи $\mathfrak{G}$.

Доведення. Нехай = $\mathfrak{H}_1$, $\mathfrak{H}_2$,…, $\mathfrak{H}_k$. буде сукупність всіх груп, спряжених з $\mathfrak{H}$. Тоді групи $A^{-1}\mathfrak{H}_1A$, $A^{-1}\mathfrak{H}_2A$,…, $A^{-1}\mathfrak{H}_kA$ являють собою ту ж сукупність, розташовану може бути в іншому порядку. Для доведення досить переконатися, що всі ці групи різні. Якщо б мало місце наприклад $A^{-1}\mathfrak{H}_iA=A^{-1}\mathfrak{H}_jA$, то, множачи зліва на A і справа на A^{-1}, отримали б $\mathfrak{H}_i=\mathfrak{H}_j$.

Нехай K – елемент, що входить до $\mathfrak{R}$. Це означає, що К одночасно входить у всі групи $\mathfrak{H}_1$, $\mathfrak{H}_2$,…, $\mathfrak{H}_k$. Тоді елемент. $A^{-1}KA$ буде одночасно входити у всі записані групи, тобто знову ж таки в $\mathfrak{H}_1$, $\mathfrak{H}_2$,…, $\mathfrak{H}_k$, а тому і в $\mathfrak{R}$. Таким чином, будь-яке перетворення А групи $\mathfrak{G}$ не виводить елементів групи $\mathfrak{R}$ з $\mathfrak{R}$, тобто $\mathfrak{R}$ є нормальний дільник групи $\mathfrak{G}$, що і треба було довести.

РОЗДІЛ 4.
КІЛЬЦЯ, ПОЛЯ ТА ЇХ ВЛАСТИВОСТІ

4.1. Означення кільця, приклади кілець.

Другою досить важливою алгебраїчною структурою, поряд з групою, є кільце.

Нехай в непорожній множині K визначені дві бінарні операції – додавання і множення.

Означення. Непорожня множина K, в якій визначені операції додавання і множення, називається кільцем, якщо виконуються такі умови:

1. Множина К є абелевою групою відносно додавання.

2. Операція множення асоціативна, тобто

$$\overset{\forall}{a,b,c} \in K[(ab)c = a(bc)]$$

3. Операція множення дистрибутивна відносно операції додавання, тобто:

$$\overset{\forall}{a,b,c} \in K[(a+b)c = ac + bc;\ c(a+b) = ca + cb].$$

Умови 1 – 3 називають аксіомами, що визначають кільце, або, для скорочення, аксіомами кільця.

Якщо операція множення в кільці К комутативна, то кільце називають комутативним.

Приклади кілець, що наводяться нижче, свідчать про те, що система аксіом кільця несуперечлива.

Приклади:

1. Множина цілих чисел Z є комутативне кільце відносно визначених у ній операцій додавання і множення. Справді, множина Z є абельова група по додаванню, операція множення чисел, як відомо, асоціативна, комутативна і дистрибутивна відносно операції додавання.

2. Множина парних чисел є комутативне кільце відносно операцій додавання і множення чисел. Справді, ця множина є абельова група по додаванню, в ній визначена операція множення: добуток парних чисел є парне число, причому операція множення асоціативна, комутативна і дистрибутивна відносно операції додавання.

3. Множина раціональних чисел Q є комутативне кільце відносно визначених у ній операцій додавання і множення. Читач доведе це самостійно.

4. Множина R всіх дійсних чисел, очевидно, також є кільце відносно визначених у ній операцій додавання і множення.

5. Множина K всіх чисел виду $a+b\sqrt{2}$, де a і b – будь-які раціональні числа, є комутативне кільце. Справді, які б ми не взяли числа $a_1+b_1\sqrt{2}$ і

$a_2+b_2\sqrt{2}$ з множини K, їх сума $(a_1+b_1\sqrt{2})+(a_2+b_2\sqrt{2})=(a_1+a_2)+(b_1+b_2)\sqrt{2}$, добуток $(a_1+b_1\sqrt{2})\cdot(a_2+b_2\sqrt{2})=(a_1a_2+2b_1b_2)+(a_1b_2+b_1a_2)\sqrt{2}$ і різниця $(a_1+b_1\sqrt{2})-(a_2+b_2\sqrt{2})=(a_1-a_2)+(b_1-b_2)\sqrt{2}$ є числа виду $a+b\sqrt{2}$ тобто належать до множини K. Отже, в множині K визначені операції додавання та множення і здійсненна обернена додаванню операція віднімання. Оскільки операції додавання і множення дійсних чисел асоціативні й комутативні. а елементи множини K є дійсні числа, то операції додавання і множення елементів множини К також асоціативні й комутативні. З цієї ж причини в множині K операція множення дистрибутивна відносно операції додавання. Отже, множина є комутативне кільце.

До цього кільця належать, зокрема, всі раціональні числа (при b = 0), а також число $\sqrt{2}$ (при a = 0, b=1). В цьому прикладі замість числа $\sqrt{2}$ можна було взяти $\sqrt{2}$, $\sqrt{5}$, $\sqrt{2}$ і інші.

6. Множина, що складається з одного числа 0, очевидно, є кільце. Це кільце називають нульовим.

Зауважимо, що множина натуральних чисел, множина цілих непарних чисел, а також множина додатних раціональних чисел не є кільцями відносно операції додавання й множення, бо в кожній з них нездійсненна операція віднімання. У всіх наведених прикладах елементами кільця є числа. Кільце, елементами якого є числа, називають числовим кільцем. Кільце називають скінченним, якщо множина його елементів скінченна.

Наведемо ще один приклад кільця, причому скінченного, елементами якого є не числа, а об'єкти іншої природи. Нехай n— довільне відмінне від 1 натуральне число.

Означення. Цілі числа a і b називають конгруентними за модулем n і записують $a\equiv b\ (mod\ n)$, якщо при діленні їх на п дістають одну й ту саму остачу, тобто якщо різниця цих чисел націло ділиться на n. Так, $5 \equiv 8 \pmod 3$, $10 \equiv 40 \pmod 6$.

Позначимо символом C_r множину всіх цілих чисел, при діленні кожного з яких на n дістаємо остачу r (r — ціле число, що задовольняє умови $0 \le r \le n-1$), і називатимемо її класом конгруентних між собою за модулем n цілих чисел. Кільце всіх цілих чисел, очевидно, можна розбити на n класів $C_0, C_1, C_2, \dots, C_{n-2}, C_{n-1}$ конгруентних між собою за модулем n цілих чисел, причому ці класи не перетинаються. У множині класів:

$$C_0, C_1, C_2, \dots, C_{n-2}, C_{n-1}, \tag{4.1}$$

слід визначити операції додавання і множення.

Нехай C_k і C_l – два будь-які (не обов'язково різні) класи з множини (4.1). Числа класу C_k є числа виду $nq+k$, а числа класу C_l — числа виду $ng+l$, де q і g – деякі цілі числа. Оскільки:

$$(nq+k)+(ng+l)=\begin{cases} n(q+g)+(k+l) & при\ k+l<n \\ n(q+g+1)+(k+l-n) & при\ k+l\geq n \end{cases},$$

тоді сума будь-якого числа з C_k і будь-якого числа з C_l є число одного й того ж класу - класу C_{k+l} при $k + l< n$ і класу C_{k+l-n} при $k+1\geq n$.

Добуток будь-якого числа з C_k на будь-яке число з C_l також є число одного й того ж класу— класу C_r, де r — остача від ділення добутку kl на n.

Справді,

$$(nq + k)+(ng+l)=n(nqg+gk+ql)+kl=n(nqg+gk+ql)+ns+r=$$
$$=n(nqg+gk+ql+s)+r=nj+r,$$

де s ціле і

$$kl=ns+r\ (0\leq r\leq n-1),\ j=nqg+gk+gl+s.$$

Тому слід операції додавання і множення в множині (4.1) визначити так.

Означення. Сумою $C_k + C_l$ класів C_k і C_l називатимемо клас, до якого належить сума будь-якого числа з C_k і будь-якого числа з C_l, а добутком C_kC_l цих класів називатимемо клас, до якого належить добуток будь-якого числа з C_k на будь-яке число з C_l, тобто:

$$C_k + C_l=\begin{cases} C_{k+1} & при\ k+l<n \\ C_{k+l-n} & при\ k+l\geq n \end{cases}.$$

$$C_k\cdot C_l=C_r,\ kl=ns+r,\ 0\leq r\leq n-1,$$

Так визначені операції додавання і множення класів множини (0) асоціативні, комутативні і пов'язані законом дистрибутивності. Це випливає із здійснення законів асоціативності, комутативності і дистрибутивності для операції додавання та множення в кільці цілих чисел й того зв'язку між операціями над цілими числами і над класами множини (1), про. який ішла мова вище, і може також бути доведено безпосередньою перевіркою.

Клас C_0, що складається з чисел, які націло діляться на n, відіграє, очевидно, в множині класів (1) роль нульового елемента. Оскільки $C_k + C_{n-k} = C_0$, то клас C_{n-k} є протилежним для класу C_k і, отже, в множині класів (1) для кожного її класу міститься і протилежний клас.

З вищевикладеного випливає, що множина класів (1) є абельова група відносно додавання, в якій визначена операція множення, причому ця операція асоціативна, комутативна і дистрибутивна відносно операції додавання. Отже, множина класів (1) є комутативне кільце. Позначатимемо його символом Z_n.

4.1.1. Елементарні відомості про кільця.

Розглянемо насамперед деякі найпростіші властивості кілець, що випливають безпосередньо з означення кільця. Нагадаємо, що множина

елементів будь-якого кільця K є абельова група відносно додавання. її називають адитивною групою кільця K.

Оскільки будь-яке кільце є абельова група відносно додавання, то всі означення, введені нами для адитивних абельових груп, і всі твердження, доведені для цих груп, повністю переносяться на кільця. Для кілець справедливі наступні твердження:

1. У кожному кільці K сума n елементів кільця не залежить від способу розставляння дужок, тобто всі суми, які дістаємо при різних розставляннях дужок, рівні між собою. У кожному кільці K сума будь-яких його елементів не залежить від порядку доданків.

2. У кожному кільці K рівні доданки в обох частинах рівності можна відкидати, якщо:

$$a + b_1 = a + b_2, \text{ то } b_1 = b_2.$$

3. У кожному кільці K існує, і притому тільки один, нульовий елемент 0 відносно операції додавання, який називають нулем кільця.

4. У кожному кільці K для будь-якого його елемента a існує(і притому тільки один), протилежний елемент $-a$.

5. У кожному кільці K для будь-яких його елементів $a_1, a_2, \ldots, a_n$ здійснюється рівність:

$$-(a_1 + a_2 + \ldots + a_n) = (-a_1) + (-a_2) + \ldots + (-a_n).$$

6. У кожному кільці K для будь-якого його елемента a і довільного натурального числа n здійснюється рівність:

$$n(-a) = -(na).$$

7. У кожному кільці K містяться кратні na (n – довільне ціле число) будь-якого елемента a. Для будь-яких елементів a і b кільця K і довільних цілих чисел m і n справджуються рівності:

$$(m + n)a = ma + na,$$
$$n(a + b) = na + nb,$$
$$m(na) = (mn)a.$$

Оскільки будь-яке кільце K є абельова група по додаванню, то для будь-яких елементів a і b кільця K існує тільки один розв'язок рівняння:

$$a + x = b.$$

Цей розв'язок називають різницею елементів a та b і позначають символом $b-a$. Але розв'язком рівняння $a + x = b$, очевидно, є $b + (-a)$. Отже:

$$b - a = b + (-a).$$

З асоціативності дії множення випливають наступні властивості кільця:

1. У кожному кільці K добуток будь-яких n його елементів не залежить від способу розставляння дужок, тобто всі добутки, які дістаємо при різних розставляннях дужок, рівні між собою.

2. Кожне кільце K містить цілі додатні степені a^n будь-якого його елемента a, причому при довільних цілих додатних m і n здійснюються такі рівності:

$$a^m \cdot a^n = a^n \cdot a^m = a^{m+n},\ (a^m)^n = (a^n)^m = a^{mn}.$$

Єдиною умовою в означенні кільця, що пов'язує дії додавання і множення, є дистрибутивність множення відносно додавання.

З дистрибутивності множення відносно додавання безпосередньо випливає справедливість таких тверджень.

Теорема. У кожному кільці K дистрибутивні закони справедливі для довільного скінченного числа доданків, тобто:

$$(a_1 + a_2 + \ldots + a_n)\, b = a_1 b + a_2 b + \ldots + a_n b, \tag{4.2}$$

$$b\,(a_1 + a_2 + \ldots + a_n) = b a_1 + b a_2 + \ldots + b a_n\,, \tag{4.3}$$

для будь-якого натурального числа n.

Доведення. Оскільки доведення справедливості обох цих рівностей аналогічне, то треба довести правильність лише першої з них.

Для $n = 2$ рівність (4.2) справедлива. Припустимо, що ця рівність правильна для $n = m$, і доведемо, що тоді вона справджується і для $n = m + 1$. Нехай

$$(a_1 + a_2 + \ldots + a_m)\, b = a_1 b + a_2 b + \ldots + a_m b,$$

Тоді

$$\begin{aligned}(a_1 + a_2 + \ldots + a_m)\, b &= [(a_1 + a_2 + \ldots + a_m) + a_{m+1}]b = \\ &= (a_1 + a_2 + \ldots + a_m)b + a_{m+1}b = \\ &= a_1 b + a_2 b + \ldots + a_m b + a_{m+1}b.\end{aligned}$$

Отже, за принципом математичної індукції, рівність (4.2) справджується для будь-якого натурального n.

Теорема. У кожному кільці K справедливе звичайне правило множення суми на суму (але без зміни порядку множників), тобто

$$\begin{aligned}&(a_1 + a_2 + \ldots + a_m)(b_1 + b_2 + \ldots + b_n) = \\ &= a_1 b + a_2 b + \ldots + a_m b_1 + a_1 b_1 + a_2 b_2 + \ldots + a_m b_2 + \ldots + \\ &+ a_1 b_n + a_2 b_n + \ldots + a_m b_n.\end{aligned}$$

Доведення. Справді, за попередньою теоремою,

$$\begin{aligned}&(a_1 + a_2 + \ldots + a_m)(b_1 + b_2 + \ldots + b_n) = \\ &= (a_1 + a_2 + \ldots + a_m)b_1 + (a_1 + a_2 + \ldots + a_m)b_2 + \ldots + \\ &+ (a_1 + a_2 + \ldots + a_m)b_n = a_1 b + a_2 b + \ldots + a_m b_1 + a_1 b_1 + a_2 b_2 + \ldots + \\ &a_m b_2 + \ldots + \\ &+ a_1 b_n + a_2 b_n + \ldots + a_m b_n\end{aligned}$$

Теорема. У кожному кільці K операція множення дистрибутивна відносно операції віднімання, тобто

$$\overset{\forall}{a,b,c} \in K[(a-b)c = ac - bc;\ c(a-b) = ca - cb]\,. \tag{4.4}$$

Доведення. Справді, $b + (a - b) = a$.

Тому $[b + (a - b)]\ c = ac$. Звідси, за законом дистрибутивності для множення і додавання, $bc + (a - b)\ c = ac$. Отже, за означенням різниці,

$(a — b)c = ac— bc$.

Аналогічно доводиться і справедливість рівності $c\ (a - b) = ca - cb$.

Теорема. У кожному кільці K добуток будь-якого його елемента a на 0 і 0 на елемент a дорівнює 0, тобто:

$$\overset{\forall}{a \in K}[a \cdot 0 = 0 \ \ a = 0].$$

Доведення. Справді, $b + 0 = b$. Отже, $0 = b - b$. Тому

$$a \cdot 0 = a\ (b- b) = ab - ab = 0,$$
$$0 \cdot a = (b - b)\ a = ba - ba = 0.$$

Теорема. У кожному кільці K для будь-яких його елементів a та b справедливі рівності:

$$(-a)\ b = -ab,\ a\ (-b) = -ab,\ (-a)\ (-b) = ab,$$

тобто справедливі звичайні «правила знаків».

Доведення. Справді, $ab + (-a)b = [a + (-a)]b = 0 \cdot b = 0$.

Таким чином, $(-a)\ b$ є елемент, протилежний елементу ab і, отже, $(-a)\ b = -ab$. Аналогічно можна довести, що $a(-b) = -ab$.

Звідси, в свою чергу, випливає, що $(-a)\ (-b) = -\ [a\ (-b)] = -(-ab) = ab$, оскільки протилежним для $-ab$ є елемент ab. З вищевикладеного ми можемо зробити висновок, що чимало відомих нам властивостей дій над числами зберігаються і для операцій, визначених у будь-якому кільці K. І це цілком закономірно, адже ці властивості випливають безпосередньо з аксіом, що визначають кільце, а кожна з множин усіх цілих, усіх раціональних, усіх дійсних чисел є кільце.

Однак не слід думати, що всяка властивість дій додавання і множення чисел зберігається для алгебраїчних операцій у будь-якому кільці. Ті властивості додавання і множення чисел, які не є наслідками з аксіом кільця, в довільному кільці K можуть не мати місця. Так, наприклад, для чисел правильне таке твердження: якщо добуток двох чисел дорівнює нулю, то принаймні один з множників дорівнює нулю. Але це твердження не буде правильне для будь-якого кільця. В деяких кільцях є такі пари відмінних від нуля елементів a і b, добуток яких дорівнює нулю, тобто $a \neq 0$, $b \neq 0$, а $ab = 0$. Так, якщо п є складене натуральне число, наприклад $n = p \cdot q$, то в кільці $\mathbf{Z_n}$ (в якому нулем є клас C_0) кожен клас C_p і C_q відмінний від нуля, а їх добуток дорівнює нулю: $C_p \cdot C_q = C_0$.

Означення. Відмінні від нуля елементи a і b кільця K, добуток яких дорівнює нулю, називаються дільниками нуля. Для елементів будь-якого кільця K справедливим буде твердження.

Теорема . Якщо елемент a кільця K відмінний від нуля і не є дільником нуля, то з рівності $ab = ac$ випливає, що $b = c$.

Доведення. Справді, з рівності $ab = ac$ випливає рівність $ab-ac = 0$, тобто $a(b-c)=0$. Але оскільки $a \neq 0$ і не є дільником нуля, то $b - c = 0$ і, отже, $b = c$. Теорему доведено.

Означення. Комутативне кільце K, яке не має дільників нуля, називають областю цілісності.

Кожне числове кільце, очевидно, є областю цілісності. Якщо K - область цілісності і a – елемент цієї області, відмінний від нуля, то за теоремою 6, з рівності $ab = ac$ випливає, що $b = c$.

Зауважимо, що з означення кільця не випливає існування або відсутність у кільці одиничного відносно множення елемента. Але якщо в кільці існує одиничний відносно множення елемент, то він єдиний. Його називають одиницею кільця.

Кільце K, в якому є одиниця, називають кільцем з одиницею.

Наведені вище приклади кілець показують, що іноді частина K_1 елементів кільця K сама є кільцем відносно алгебраїчних операцій, визначених у K. Тоді K_1 називають підкільцем кільця K. Точніше:

Означення. Підмножина K_1 кільця K називається підкільцем кільця K, якщо K_1 є кільцем відносно операцій додавання і множення, визначених у кільці K.

Так, кільце парних чисел є підкільцем кільця цілих чисел, а останнє, в свою чергу, є підкільцем кільця раціональних чисел. Кільце раціональних чисел і кільце чисел вигляду $(a + b\sqrt{2})$, де a і b — будь-які раціональні числа, є підкільцями кільця дійсних чисел.

У кожному кільці K, очевидно, є такі підкільця: само кільце K і нульове підкільце, яке складається лише з елемента 0. Ці підкільця називають тривіальними.

Для того щоб з'ясувати, чи є дана непорожня підмножина K_1 кільця K його підкільцем, зручно користуватися такою теоремою.

Теорема. Для того щоб непорожня підмножина K_1 кільця K була його підкільцем, необхідно й достатньо, щоб сума, різниця й добуток будь-яких двох елементів підмножини K_1 належали до K_1

Доведення. Доведемо спочатку необхідність умови. Припустимо, що K_1 – підкільце кільця K. Тоді додавання (множення) елементів підкільця K_1, за означенням підкільця, збігається з додаванням (множенням) їх як елементів кільця K. Але оскільки для операції додавання обернена їй операція віднімання визначається однозначно, то і віднімання елементів підкільця K_1 збігається з відніманням їх як елементів кільця K. Тому сума, різниця і добуток будь-яких двох елементів із K_1 належать до K_1, бо в противному разі принаймні одна з операцій додавання, віднімання, множення була б нездійсненна в K_1 і, отже, підмножина K_1 не була б

підкільцем. Доведемо тепер достатність умови. Припустиш, що підмножина K_1 задовольняє умови теореми.

Оскільки сума і добуток (що визначені в K) будь-яких двох елементів a_1 і b_1 з K_1 належать до K_1 то приймемо їх за результат додавання і множення елементів a_1 та b_1 в підмножині K_1 і, таким чином, введемо в K_1 операції додавання і множення. Визначена таким способом операція додавання в K_1 асоціативна і комутативна, а операція множення – асоціативна і дистрибутивна відносно операції додавання. Справді, оскільки

$$\overset{\forall}{a,b,c} \in K[(a+b)+c = a+(b+c)\Lambda a+b = b+a\Lambda(ab)c = a(bc)\Lambda(a+b)c =$$
$$= ac+bc\Lambda c(a+b) = ca+cb]$$

то, зокрема, й:

$$\overset{\forall}{a_1,b_1,c_1} \in K_1[(a_1+b_1)+c_1 = a_1+(b_1+c_1)\Lambda a_1+b_1 = b_1+a_1\Lambda(a_1b_1)c_1 =$$
$$= a_1(b_1c_1)\Lambda(a_1+b_1)c_1 = a_1c_1+b_1c_1\Lambda c_1(a_1+b_1) = c_1a_1+c_1b_1].$$

У підмножині K_1 здійсненна також операція віднімання. Справді, нехай a_1 і b_1 – довільні елементи з K_1. Рівняння $a_1 + x_1 = b_1$ має в кільці K єдиний розв'язок $b_1 – a_1$ який за умовою належить до K_1. Оскільки додавання елементів у K збігається з додаванням їх як елементів кільця K, то і в K_1 справджується рівність $a_1 + (b_1 – a_1) = b_1$. Отже, $b_1 – a_1$ є розв'язком рівняння $a_1 + x_1 = b_1$ й у підмножині K_1, причому єдиним, бо в противному разі цей розв'язок не був би єдиним і в кільці K. З викладеного випливає, що K є кільце відносно операції додавання і множення, визначених у кільці K, тобто є підкільце кільця K.

4.2. Поля та їх властивості.

4.2.1. Означення поля. Приклади полів.

В кожному кільці для операції додавання здійсненна обернена операція віднімання. Про здійсненність же ділення – операції, оберненої множенню, в означенні кільця не говориться нічого. Наведені в п. 4.1 приклади кілець, показують, що по відношенню до операції ділення різні кільця мають різні властивості. Так, у кільці парних чисел ділення здійсненне у виняткових випадках; в цьому кільці немає жодного елемента, на який ділилися б усі інші його елементи. У кільці цілих чисел дія ділення також здійсненна лише у виняткових випадках, але всі елементи цього кільця діляться на 1 і -1. А в кільці раціональних чисел дія ділення здійсненна завжди, крім ділення на нуль. Це саме можна сказати про кільце дійсних чисел. Зауважимо, що ділення на нуль неможливе в жодному кільці: поділити елемент a на нуль – це означає знайти в кільці такий елемент b, для

якого $0 \cdot b = a$, а це при $a \neq 0$ неможливо, оскільки для будь-якого елемента c кільця $0 \cdot c = 8$.

Важливу роль у математиці відіграють комутативні кільця, в яких здійсненна дія ділення, крім ділення на нуль. їх називають полями. Точніше:

Означення. Комутативне кільце P називається полем, якщо воно має принаймні один елемент, відмінний від нуля, і якщо в ньому в усіх випадках здійсненна дія ділення, крім ділення на нуль, тобто якщо для будь-яких його елементів a і b, з яких $a \neq 0$, в ньому міститься, і притому тільки один, такий елемент q, для якого $ag = b$.

Елемент q називають часткою елементів b і a і записують: $q = \frac{b}{a}$.

Вимоги, що входять до означення поля, називають аксіомами поля.

Полем, наприклад, є кільце раціональних чисел Q, кільце дійсних чисел R. Полем є також кільце дійсних чисел виду $a + b\sqrt{2}$, де a і b – будь-які раціональні числа. Справді, в цьому кільці завжди здійсненне ділення, крім ділення на нуль. Щоб довести це, розглянемо частку чисел $a_1 + b_1\sqrt{2}$ і $a_2 + b_2\sqrt{2}$ причому останнє число відмінне від нуля (тобто a_2 і b_2 не можуть одночасно дорівнювати нулю). Отримується наступне співвідношення:

$$\frac{a_1 + b_1\sqrt{2}}{a_2 + b_2\sqrt{2}} = \frac{\left(a_1 + b_1\sqrt{2}\right)\left(a_2 + b_2\sqrt{2}\right)}{a_2^2 - 2b_2^2} = \frac{a_1a_2 - 2b_1b_2}{a_2^2 - 2b_2^2} + \frac{a_2b_1 - a_1b_2}{a_2^2 - 2b_2^2}\sqrt{2} = c_1 + c_2\sqrt{2},$$

$$c_1 = \frac{a_1a_2 - 2b_1b_2}{a_2^2 - 2b_2^2}, \quad c_2 = \frac{a_2b_1 - a_1b_2}{a_2^2 - 2b_2^2}$$

Оскільки c_1 і c_2 – раціональні числа, то частка двох чисел з розглядуваного кільця належить цьому ж кільцю.

Зауважимо, що вирази для c_1 і c_2 завжди мають смисл (якщо a_2 і b_2., одночасно не дорівнюють нулю), оскільки $a_2^2 - 2b_2^2 \neq 0$. Справді, нехай $a_2^2 - 2b_2^2 = 0$; тоді $\left(a_2 / b_2\right)^2 = 2$, а це суперечить відомому факту, що квадрат жодного раціонального числа не може дорівнювати 2. Отже, кільце чисел виду $a + b\sqrt{2}$, справді є полем.

Ці поля, як і всі інші, елементами яких є числа, називають числовими полями.

Приклад нечислового поля. Нехай комутативне кільце Z_p класів конгруентних за модулем р чисел, де p – деяке просте число. Слід показати, що кільце Z_p є поле. Справді, в кільці Z_p, очевидно, є елементи, відмінні від нульового елемента C_0. Потрібно довести, що в кільці Z_p здійсненна операція ділення, крім ділення на нульовий елемент C_0. Нехай $C_к$ і C_m – будь-які класи з кільця Z_p, причому $C_k \neq C_0$. Клас C_m можна поділити на клас Cm тобто, що в кільці Z_p міститься, і притому лише один, клас C_l, який

задовольняє умову $C_l \cdot C_к = C_m$. Якщо $C_m = C_0$, то й $C_l = C_0$. Припускається, що $C_m \neq C_0$. Тоді розглядається ряд натуральних чисел:

$$k,\ 2k,\ 3k,\ \ldots,\ (p-1)\,k \qquad (4.5)$$

Жодне з цих чисел не ділиться на p, бо добуток двох натуральних чисел, менших, ніж просте число p, не може ділитися на p, і тому жодне з них не міститься в класі C_0.

Крім того, ніякі два числа sk і $tk (t > s)$ ряду (1) не можуть міститись в одному класі, бо тоді їх різниця $tk - sk = (t - s)\,k$ повинна була б ділитись на p, чого, як зазначено вище, не може бути. Отже, в кожному з ненульових класів кільця Z_p, зокрема й у класі C_m, міститься одне і тільки одне число ряду (4.5).

Нехай у класі C_m міститься число $lk\left(l \le l \le p-1\right)$ Тоді $C_l \cdot C_k = C_m$ і, отже, клас C_l і є шуканим класом, причому, як випливає з вищевикладеного, це єдиний клас, який задовольняє умову $C_l \cdot C_k = C_m$.

Отже, кільце Z_p є поле, причому скінченне: воно складається з p елементів. Звідси випливає, що існує нескінченне число різних скінченних полів: Z_2, Z_3, Z_5, Z_7, ... і т. д.

4.2.2. Властивості полів.

Оскільки кожне поле є комутативним кільцем, то всі розглянуті властивості кілець і твердження, що стосуються кілець, правильні також і для будь-якого поля. Властивості полів, що випливають із здійсненності дії ділення. Відповідно до означення, елемент поля P, добуток якого на будь-який елемент $a \in P$ дорівнює a, називається одиничним елементом, або одиницею поля, і позначається символом e.

Теорема. У кожному полі P існує, і притому тільки одна, одиниця.

Доведення. За означенням, у полі Р існує принаймні один елемент $a \neq 0$.

Оскільки в полі P здійснена дія ділення, крім ділення на нуль, то в ньому існує частка – $\frac{a}{a}$. Позначимо її символом e. Тоді $ae = ea = a$. Нехай b – будь-який елемент поля P. Покажемо, що $be = eb = b$. Справді, у полі P існує елемент q, що $qa = aq = q$. Отже, $eb = be = (qa)\,e = q\,(ae) = qa = b$, тобто $eb = be = b$ і тому e – одиничний елемент поля P. Отже, одиничний елемент у полі P існує тільки один. Теорему доведено.

Елемент поля P, добуток якого на елемент $a \in P$ дорівнює e, називається оберненим елементом a і позначається символом a^{-1}.

Теорема. У кожному полі P для будь-якого його елемента a, відмінного від нуля, існує, і притому тільки один, обернений елемент a^{-1}.

Доведення. Справді, у полі P існує частка $\frac{e}{a}$ елементів e і a. Її можна позначити символом a^{-1}. Тоді $aa^{-1} = a^{-1}a = e$. Цим існування елемента a^{-1}, оберненого елементу a, доведено.

Для елемента $a \in P$ у полі P існує лиш один обернений елемент. Теорему доведено.

Використовуючи наявність у полі P обернених елементів, частку $\frac{a}{b}$ елементів b і a поля P можна записати так: $\frac{a}{b} = b\,a^{-1}$.

Теорема. Жодне поле P не має дільників нуля: $a \cdot b = 0 \Rightarrow a = 0 \vee b = 0$

Доведення. Справді, нехай $a \cdot b = 0$ і $a \neq 0$. Тоді, помноживши зліва обидві частини рівності $a \cdot b = 0$ на a^{-1}, матимемо $a^{-1}(ab) = a^{-1}0$, тобто $(a^{-1}a)b = 0$ і, отже, $b = 0$.

З останньої теореми випливає, що добуток відмінних від нуля елементів a і b поля P є також відмінний від нуля елемент цього поля. Отже, операція множення елементів поля P визначена також і в множині G відмінних від нуля елементів цього поля. З доведених теорем випливає справедливість такої

Теорема. В кожному полі множина G елементів, відмінних від нуля, є абельова група відносно множення, вона називається мультиплікативною групою поля P.

Доведення. Справді, в множині G визначена операція множення. Оскільки $G \in P$, то ця операція асоціативна і комутативна. Одиничний елемент e поля P відмінний від нуля, бо для кожного елемента a з поля P $a0 = 0$, а $ae = a$. Тому e міститься в множині G. Так само, для кожного відмінного від нуля елемента a поля P обернений елемент a^{-1} відмінний від нуля, бо $a0 = 0$, а $aa^{-1} = e$, і тому a^{-1} міститься в множині G.

Отже, за другим означенням групи, G є абельова група відносно множення. Теорему доведено.

Оскільки множина відмінних від нуля елементів поля P є абельова група відносно множення, то всі властивості мультиплікативної групи має і поле P. Зокрема в кожному полі P можна скорочувати на множник, відмінний від нуля: якщо $ab = ac$ і $a \neq в$, то $b = c$.

У кожному полі P зберігаються звичайні правила дій над степенями з цілими показниками, тобто при будь-яких цілих показниках m і n для будь-якого елемента a поля P справедливі такі рівності:

$$a^m \cdot a^n = a^{m+n},$$
$$(a^m)^n = a^{mn}.$$

Доведемо тепер, що в будь-якому полі зберігаються також і звичайні правила дій над частками (дробами), а саме:

а) якщо $a \neq 0$ і $a_1 \neq 0$, то $\frac{b}{a} = \frac{b_1}{a_1}$ - тоді і тільки тоді, коли $ba_1 = ab_1$;

б) $\overset{\forall}{a, a_1, b, b_1} \in P\left[a \neq 0 \Lambda a_1 \neq 0 \Rightarrow \frac{b}{a} \pm \frac{b_1}{a_1} = \frac{ba_1 \pm ab_1}{aa_1} \right]$;

в) $\overset{\forall}{a, a_1, b, b_1} \in P\left[a \neq 0 \Lambda a_1 \neq 0 \Rightarrow \frac{b}{a} \cdot \frac{b_1}{a_1} = \frac{b \cdot b_1}{aa_1} \right]$;

г) $\overset{\forall}{a, a_1, b, b_1} \in P\left[a \neq 0 \Lambda a_1 \neq 0 \Rightarrow \frac{b}{a} : \frac{b_1}{a_1} = \frac{ba_1}{ab_1} \right]$;

д) $\overset{\forall}{a, b} \in P\left[a \neq 0 \Rightarrow \frac{-b}{a} = -\frac{b}{a} \right]$.

З викладеного вище, випливає, що в будь-якому полі правильні всі твердження і формули елементарної алгебри, що ґрунтуються на правилах дій над степенями з цілими показниками і над частками (дробами).

Розглянуті нами властивості поля^ випливають з означення поля і не залежать від властивостей його елементів. Однак є властивості полів, які визначаються особливостями елементів поля.

4.2.3. Характеристика поля.

Нехай P – деяке числове поле, а Z_p – поле класів C_0, C_1, C_2, ..., C_{p-1} цілих чисел, конгруентних за модулем p (p – деяке просте число). Яке б ціле додатне кратне одиниці поля Р не бралося, воно, очевидно, ніколи не дорівнюватиме нулю.

Всі такі кратні, тобто всі натуральні числа, відмінні одне від одного. Якщо ж брати різні цілі додатні кратні одиниці поля Z_p, тобто класу C_1, то деякі з них дорівнюватимуть нулю. Очевидно, що $pC_1 = C_0$, $2pC_1 = C_0$ і взагалі при будь-якому натуральному k слідує рівність $kpC_1 = C_0$.

Нехай P – деяке поле. Робиться припущення, що будь-яке ціле додатне кратне одиниці є поля P відмінне від нуля. Тоді і всяке ціле від'ємне кратне є також відмінне від нуля, бо якщо елемент поля – $ne (n > 0)$ дорівнює нулю, то й протилежний йому елемент пе також дорівнює нулю. Отже, тільки нульове кратне є дорівнює нулю, тобто тільки при $n = 0$ справджується рівність $ne = 0$. В цьому випадку говорять, що поле P має характеристику нуль, або що Р є поле характеристики нуль.

Тепер припускається, що деяке ціле додатне кратне e, наприклад se, дорівнює нулю. Тоді і всяке ms – кратне одиниці (де m – довільне ціле число), також дорівнює нулю, бо $(ms)e = m(se) = m0 = 0$

Серед усіх цілих додатних кратних одиниці e, що дорівнюють нулю, очевидно, є таке, в якого коефіцієнт кратності найменший. Нехай таким коефіцієнтом є натуральне число p. Отже, $pe = 0$, і немає такого натурального числа k, меншого ніж p, що $ke = 0$. В цьому випадку P називається полем скінченної характеристики, а саме характеристики p. Таким чином, поняття характеристики поля можна означити так.

Означення. Характеристикою поля P називається число нуль, якщо $ne = 0$ лише при $n = 0$; характеристикою поля P називається натуральне число p, якщо $pe = 0$ і немає такого натурального числа k, меншого ніж p, що $ke = 0$.

Всі числові поля є поля характеристики 0. Всі ж скінченні поля є прикладами полів скінченної характеристики. Справді, якщо поле P – скінченне, то серед усіх цілих додатних кратних одиниці цього поля обов'язково будуть кратні, рівні між собою, бо в противному разі поле P було б нескінченним. Нехай $ke = le$, де k і l — деякі натуральні числа, причому $k > l$. Тоді $(k — l)\, e = 0$ і, отже, P є поле скінченної характеристики.

Будь-яке просте число p, очевидно, є характеристикою поля $\boldsymbol{Z_p}$. Але жодне складене число не може бути характеристикою поля. Інакше кажучи, не існує полів, характеристиками яких були б складені числа, оскільки правильне таке твердження:

Теорема. Якщо поле P має характеристику p, то число p — просте.

Доведення. Нехай p - складене число, і нехай $p = st$, де $s < p$, $t < p$. Тоді:

$(se)(te)=(e+e+\ldots+e)\cdot(e+e+\ldots+e)=ee+ee+\ldots+ee=e+e+\ldots+e=ste=pe=0$

тобто $(se) \cdot (te) = 0$.

Оскільки в полі не може бути дільників нуля, то з рівності $(se)(te) = 0$ випливає, що або $se = 0$, або $te = 0$, а це суперечить умові, що поле P має характеристику p. Отже, припущення, що p – складене число, приводить до суперечності, тому воно неправильне. Теорему доведено.

Тепер потрібно розглянути деякі властивості поля характеристики 0 і характеристики p.

Теорема. Якщо P є поле характеристики 0, a – будь-який відмінний від нуля елемент цього поля і n – довільне відмінне від нуля ціле число, то $na \neq 0$.

Доведення. Нехай n – будь-яке натуральне число. Тоді

$na = a + a + \ldots + a = a(e + e + \ldots + e) = a\,(ne)$.

Нехай $na = 0$, тобто $a(ne) = 0$. Оскільки в полі не може бути дільників нуля і за умовою $a \neq 0$, то з рівності $a\,(ne) = 0$ випливає, що $ne = 0$, чого не може бути. Таким чином, припущення, що $na = 0$ неправильне і, отже, при будь-якому натуральному n маємо $na \neq 0$; $na \neq 0$ також і при будь-якому цілому від'ємному n, бо якби елемент $na\ (n < 0)$ поля P дорівнював нулю, то

й протилежний йому елемент (-*n*) а також дорівнював би нулю, чого, за доведеним вище, не може бути.

Теорема. Якщо *P* – поле характеристики *p*, то для будь-якого елемента *a* цього поля справджується рівність *pa = 0.*

Доведення. Справді,

pa = (a + a + ...+ a) = a (e + e + ... + e) = a (pe) = a · 0 = 0.

4.2.4. Підполе, розширення поля.

Аналогічно означенню підкільця означається і поняття підполя.

Означення. Підмножина *P′* поля *P* називається підполем цього поля, якщо вона сама є полем відносно алгебраїчних операцій, визначених у полі *P*. Поле *P* в цьому випадку називається розширенням поля *P′.*

Очевидно, що нуль і одиниця поля *P* міститься в підполі *P′* і є відповідно нулем та одиницею й у підполі *P*. Так, поле раціональних чисел *Q* є підполе поля дійсних чисел виду $a + b\sqrt{2}$, де *a* і *b* – будь-які раціональні числа, і поля всіх дійсних чисел *R*; поле дійсних чисел виду $a + b\sqrt{2}$ також є підполе поля дійсних чисел *R*.

Теорема. Для того щоб підмножина *P′* поля *P,* яка містить принаймні один елемент, відмінний від нуля, була підполем, необхідно й достатньо, щоб сума, різниця, добуток і частка (якщо вона існує) будь-яких двох елементів підмножини *P′* містилися в підмножині *P′.*

Для підполів будь-якого поля *P* правильне таке твердження:

Теорема. Перетин будь-якої множини підполів поля *P* також є підполе поля *P*.

Доведення. Нехай *{Ps}* (індекс *s* пробігає деяку множину значень *S*) є деяка множина підполів поля *P i* $D = \bigcap_{s \in S} P_s$ є перетин усіх підполів цієї множини.

Елементи *0* і є поля *P* входять до кожного з підполів P_s, а тому і до їх перетину *D*. Отже, *D* містить принаймні один елемент, відмінний від *0*. Нехай *a* і *b* – два довільних елементи з перетину *D*. Елементи *a* і *b*, очевидно містяться в кожному з підполів P_s. За означенням поля, елементи *a+b, a -b,* $a \cdot b$ *i* $\frac{a}{b}$ при $b \neq 0$ також містяться в кожному підполі P_s, а тому і в їх перетині *D*. Отже, згідно відповідних властивостей, *D* є підполе поля P. Теорему доведено.

4.2.5. Упорядковані кільця і поля.

В деяких кільцях і полях, поряд з бінарними операціями введено також відношення порядку 2, яке пов'язане з бінарними операціями, їх називають упорядкованими. Точніше:

Означення. Кільце (зокрема поле) К називається упорядкованим, якщо множина його елементів лінійно упорядкована й відношення порядку $a < b$ («а менше від b») задовольняють умовам:

1) $\overset{\forall}{a,b,c} \in K\left[a < b \Rightarrow a + c < b + c\right]$,

2) $\overset{\forall}{a,b,c} \in K\left[a < b \wedge c > 0 \Rightarrow ac < bc\right]$

Умова 1) називається законом монотонності додавання, а 2) законом монотонності множення.

Якщо елемент a упорядкованого кільця K менший від елемента b цього кільця, тобто $a < b$, то говорять, що елемент b більший від елемента a, і записують $b > a$.

Прикладами упорядкованих кілець є кільце цілих чисел, кільце парних чисел, а прикладом упорядкованого поля – поле раціональних чисел, якщо під відношенням порядку розуміється звичайне відношення «менше» (число а менше від числа b).

4.3. Ізоморфізм алгебраїчних структур.

4.3.1. Поняття ізоморфізму.

Абстрактне поняття групи (кільця, поля) дає змогу одночасно вивчати спільні властивості багатьох різних множин з введеними у них алгебраїчними операціями – всіх множин, що є групами (кільцями, полями), і глибше досліджувати природу алгебраїчних операцій, абстрагуючись від конкретних об'єктів їх застосування.

Наявність таких спільних властивостей не виключає того, що різні групи (або кільця чи поля) можуть істотно відрізнятись між собою своєю будовою, мати різні індивідуальні особливості. Так, наприклад, мультиплікативна група, що складається з чисел 1 і – 1, скінченна, а адитивна група цілих чисел – нескінченна. Числові кільця не мають дільників нуля, а кільце Z_n класів чисел, конгруентних за модулем n, має дільники нуля, якщо n – складене число. Поле раціональних чисел – нескінченне, а поле Z_p (p – просте число) – скінченне.

Отже, хоч аксіоми групи, кільця, поля визначають суттєві спільні риси відповідних алгебраїчних структур, – вони ще не дають повної однозначної характеристики цих структур.

Проте це не означає, що кожну групу, кожне кільце або поле треба вивчати окремо. Уважно розглядаючи приклади конкретних алгебраїчних структур, можна помітити, що в деяких з них властивості цілком аналогічні.

Приклад. Нехай мультиплікативна група G складається з усіх цілих степенів числа 7:

$\ldots 7^{-n}, 7^{-(n-1)}), \ldots, 7^{-3}, 7^{-2}, 7^{-1}, 7^{0} = 1, 7, 7^{2}, \ldots, 7^{n-1}, 7n, \ldots$

і адитивна група Z всіх цілих чисел:

…-n, -(n-1), …, -3, -2, -1, 0, 1, 2, 3, … , n-1, n, …

Отже, кожному елементу *7n* групи *G* поставимо у відповідність елемент n групи *Z*, тобто степеню *7n* ставиться у відповідність його показник *n*. Цим, очевидно, буде задано взаємно однозначне відображення групи *G* на групу Z, причому таке, що добуток $7k \bullet 7l = 7k+l$ елементів *7k і 7l* групи *G* відповідає сума $k+l$ образів *k* і *l* цих елементів у групі *Z*.

Звідси випливає, що будь-яке доведене твердження про множення елементів групи *G* заміною кожного з чисел 7^n відповідним числом *n* і добутків – сумами може бути перетворене у відповідне твердження про додавання елементів групи *Z* і навпаки. Тому ці дві групи не можна вважати істотно різними. Розглянутий приклад показує, що різні конкретні алгебраїчні структури одного й того самого типу можна об'єднувати в класи подібних між собою структур з тим, щоб вивчення якогось одного представника цього класу повністю заміняло вивчення будь-якої і з структур, що ввійшли в цей клас.

Подібність двох конкретних алгебраїчних структур *M* і M_1 одного і того самого типу проявляється насамперед у можливості встановити взаємно однозначну відповідність між елементами множин *M* і M_1. Крім взаємно однозначної відповідності між елементами множин *M* і M_1 повинна існувати взаємно однозначна відповідність між алгебраїчними операціями, введеними у цих множинах. При цьому відповідність між елементами і відповідність між операціями повинні бути так пов'язані між собою, щоб результатам операцій над елементами множини *M* відповідали результати відповідних операцій над відповідними елементами множини M_1.

Таку відповідність між множинами *M* і *Mx* називають у математиці ізоморфізмом.

Нехай задано множину *M* з однією або двома введеними в ній бінарними операціями і множину M_1 з таким же числом бінарних операцій. Позначатимемо закони композиції у множині *M* символами τ і $*$, а у множині M_1 символами τ і $*$

Означення. Множини *M і* M_1 з введеними в них бінарними операціями називаються ізоморфними, якщо між елементами цих множин і введеними в них операціями можна встановити взаємно однозначні відповідності так, що коли елементу $a \in M$ відповідає елемент $a \in M_1$

елементу $b \in M$ – елемент $b \in M_1$ операції τ у множині M – операція τ у множині M_1, а операції $*$ у множині M – операція $*$ у множині M_1 то елементу $a \tau b \in M$ відповідає елемент $a \tau b \in M_1$, а елементу $a * b \in M$ – елемент $a_1 * b_1 \in M_1$.

Взаємно однозначна відповідність між множинами M і M_1 що задовольняють цю умову, називається ізоморфною відповідністю між множинами M і M_1 або ізоморфним відображенням множини M на множину M_1. Ізоморфізм множин M і M_1 символічно записують так: $M \underline{\infty} M_1$.

Кожна алгебраїчна структура M ізоморфна сама собі. Справді, тотожне відображення множини M самої на себе є ізоморфною відповідністю.

Отже, відношення ізоморфізму рефлексивне: $M \underline{\infty} M$.

Читач легко самостійно доведе, що відношення ізоморфізму симетричне (з $M \underline{\infty} M_1$ випливає $M_1 \underline{\infty} M$) і транзитивне (з $M \underline{\infty} M_1$ і $M_1 \underline{\infty} M_2$ випливає $M_1 \underline{\infty} M_2$).

Нехай M і M_1 – дві ізоморфні алгебраїчні структури. Цілком зрозуміло, що для кожного співвідношення між елементами із M (вираженого за допомогою алгебраїчних операцій, визначених у множині M) маємо таке саме співвідношення між відповідними елементами множини M (виражене за допомогою відповідних алгебраїчних операцій). Тому всі властивості елементів множини M, які випливають з таких співвідношень, матимуть і відповідні елементи множини M_1 і навпаки. Інакше кажучи, все, що може бути доведено для елементів однієї з цих множин, виходячи лише з властивостей визначених у ній алгебраїчних операцій, без використання індивідуальних властивостей її елементів, буде правильне і для відповідних елементів другої множини.

Отже, структури M і M_1 з точки зору визначених у них алгебраїчних операцій нічим не відрізняються одна від одної, вони можуть відрізнятися лише природою своїх елементів і, можливо, назвою заданих у них операцій та символікою, яка використовується для їх позначення. Тому в алгебрі ізоморфні структури вважають лише різними екземплярами однієї і тієї самої множини з певними алгебраїчними операціями і цим самим виділяють алгебраїчні операції як об'єкт вивчення. Алгебраїчні структури здебільшого вивчаються з точністю до ізоморфізму. Таким чином, алгебра вивчає переважно властивості самих алгебраїчних операцій, а не властивості тих об'єктів, над якими ці операції виконуються.

4.3.2 Ізоморфізм груп.

Нехай G і G_1 – групи з груповими операціями відповідно τ і τ.

Означення. Групи G *і* G_1 називаються ізоморфними, якщо між їхніми елементами можна встановити таку взаємно однозначну відповідність, що коли будь-яким елементам a, b *з* G відповідають елементи a_1, b_1, *з* G_1 то композиції $a \tau b$ відповідає композиція $a_1 \tau b_1$, тобто одна з них ізоморфно відображається на другу.

Так, адитивна група цілих чисел ізоморфна адитивній групі парних чисел. Справді, якщо кожному цілому числу k поставити у відповідність парне число $2k$, то отримується ізоморфне відображення першої групи на другу.

Мультиплікативна група додатних дійсних чисел ізоморфна адитивній групі всіх дійсних чисел.

Справді, поставивши у відповідність кожному додатному дійсному числу a дійсне число Ina, ми отримується взаємно однозначне відображення першої групи на другу, яке буде ізоморфним, оскільки

$In (a \cdot b) = In\ a + In\ b.$

В [2] наведено наступні твердження:

Теорема. При ізоморфному відображенні групи G на групу G_1 нейтральному елементу групи G відповідає нейтральний елемент групи G_1 і будь-якій парі взаємно симетричних елементів g *і* g' групи G відповідає пара взаємно симетричних елементів g_1 і g_1' групи G_1.

Доведення. Нехай нейтральному елементу η групи G при заданій ізоморфній відповідності відповідає елемент $g_1°$ групи G_1, а довільний елемент g_1 групи G_1 є відповідним елементу g групи G. Тоді маємо: $\eta \tau g = g \tau \eta = g$. Внаслідок ізоморфізму груп $g_1° \tau g_1 = g_{1\tau}\ g_1° = g_1$.Отже, g_1 нейтральний елемент $-\eta_1$ групи G_1.

Нехай тепер g і g' – довільна пара взаємно симетричних елементів групи G, *а* g_1 *і* g_1' відповідні їм елементи групи G_1. Тоді $g \tau g' = \eta$. Звідси, внаслідок ізоморфізму груп, $g_1 \tau g_1' = \eta_1$ і, отже, g_1 і g_1' - взаємно симетричні елементи групи G_1.

Теорема. Якщо множина F, в якій визначена бінарна операція, ізоморфна деякій групі G, то вона також є групою відносно введеної в ній бінарної операції.

Доведення. Нехай бінарна операція, введена в групі G, позначається символом τ, а в множині F – символом τ. Спочатку потрібно довести асоціативність бінарної операції τ.

Нехай при ізоморфізмі $F \infty G$ елементам f_1, f_2, f_3 множини F відповідають елементи групи g_1, g_2, g_3. Тоді елементу $f_1 \tau f_2$ відповідає елемент групи $g_1 \tau g_2$, а елементу $(f_1 \tau f_2)\ \tau f_3$ – елемент групи $(g_1 \tau g_2) \tau g_3$. Але $(g_1 \tau g_2) \tau g3 = g_1 \tau (g_2 \tau g_3)$. Отже, внаслідок взаємно однозначного характеру

відповідності, будуть рівними і елементи $(f_1 \tau f_2) \tau f_3$. та /і $f_1 \tau(f_2) \tau f_3$) множини F, що й треба було довести.

Дослівним повторенням міркувань, що проводилися при доведенні попередньої теореми, можна довести, що нейтральному елементу групи G відповідає нейтральний елемент множини F, а парі взаємно симетричних елементів g і g' групи G відповідає пара взаємно симетричних елементів множини F. Отже, в множині F є нейтральний елемент і для кожного елемента f є симетричний елемент f'. Таким чином, множина F є група.

4.3.3. Ізоморфізм кілець і полів.

Означення. Кільця (або поля) K *і* K_1 називаються ізоморфними, якщо між їхніми елементами можна встановити таку взаємно однозначну відповідність, що для будь-яких елементів a, b *з* K і відповідних їм елементів a, b, K_1 сумі $a + b$ відповідає сума $a_1 + b_1$, добутку ab відповідає добуток a_1b_1.

Оскільки будь-яке кільце є абельова група відносно додавання, то для ізоморфних кілець і полів будуть правильними твердження, що між кільцями K і K_1 встановлено ізоморфну відповідність, то при цій відповідності нулю 0 кільця K відповідає нуль 0_1 кільця K_1, взаємно протилежним елементам a і $-a$ кільця K відповідають взаємно протилежні елементи a_1 і $-a_1$ кільця K_1. Звідси випливає, що різниці $a- b = a + (-b)$ елементів кільця K відповідає різниця відповідних елементів $a_1-b_1=a_1+(-b_1)$ кільця K_1.

Якщо в кільці K є одиничний елемент є, то дослівним повторенням міркувань, які проводились при розгляді ізоморфних груп, легко довести, що одиниці є кільця K при ізоморфному відображенні відповідає одиничний елемент e_1 кільця K_1, і якщо для елемента a з K існує в кільці K обернений елемент a^{-1}, то елементу a_1 в кільці $K1$ відповідатиме елемент a_1^{-1} обернений елементу a_1. Звідси випливає, що кільце, ізоморфне полю, саме буде полем.

Справді, дія множення в кільці, ізоморфному полю, комутативна. В кільці, ізоморфному полю, міститься одиничний елемент і для кожного елемента а кільця існує обернений елемент a^{-1} і, отже, здійсненна дія ділення, крім ділення на нуль.

Очевидно також, що при ізоморфній відповідності зберігається властивість кільця не мати дільників нуля. Звідси, в свою чергу, випливає, що кільце K_1, ізоморфне області цілісності K, також є область цілісності.

Теорема. Якщо множина F, в якій визначені дві бінарні операції – додавання і множення, ізоморфна деякому кільцю K, то множина F також є кільце відносно визначених в ній операцій.

Доведення. Вище було доведено, що коли множина *F,* в якій визначена бінарна операція, ізоморфна деякій групі *G*, то множина *F* також є група відносно визначеної в ній операції.

Оскільки кільце *K* є група відносно додавання, то і множина *F* є група відносно додавання. Доведемо тепер, що операція додавання в множині *F* комутативна, а операція множення – асоціативна і дистрибутивна відносно додавання.

Справді, нехай f_1, f_2, f_3 – довільні елементи множини *F*. Згідно з ізоморфною відповідністю між *K* і *F*, в кільці *K* існують прообрази $a_1\ a_2,\ a_3$ цих елементів.

Для елементів $a_1\ a_2,\ a_3$ кільця *K* виконуються рівності:

$a_1 + a_2 = a_2 + a_1$

$(a_1a_2)\ a_3 = a_1\ (a_2a_3),$

$(a_1 + a_2)\ a_3 = a_1a_3 + a_2a_3,$

$a_3\ (a_1 + a_2) = a_3a_1 + a_3a_2.$

Тому внаслідок ізоморфізму $K \underline{\infty} F$

$f_1 + f_2 = f_2 + f_1$

$(f_1f_2)\ f_3 = f_1\ (f_2f_3),$

$(f_1 + f_2)\ f_3 = f_1f_3 + f_2f_3,$

$f_3\ (f_1 + f_2) = f_3f_1 + f_3f_2.$

які означають, що в множині F операція додавання комутативна, а операція множення – асоціативна і дистрибутивна відносно операції додавання. Цим теорему доведено.

З доведеного вище безпосередньо випливає правильність такої

Теореми. Якщо множина *F,* в якій визначені операції додавання і множення, ізоморфна деякому полю *P,* то множина *F* також є поле відносно визначених у ній операцій.

Справді, за попередньою теоремою F є кільце. Але, як доведено вище, кожне кільце, ізоморфне полю, також є поле. Отже, F є поле.

На закінчення цього розділу слід зазначити, що наведені вище означення різних алгебраїчних структур (групи, кільця, поля, бульової алгебри) є аксіоматичними означеннями: кожна структура означалася за допомогою певної системи аксіом. Відносно кожної з цих систем аксіом постає питання: чи задовольняє вона вимогам несуперечливості, незалежності і повноти.

Щодо вимоги несуперечливості зауважимо лише, що наявність прикладів, груп, кілець і полів, елементами яких є раціональні числа, тобто існування інтерпретації цих систем аксіом у термінах теорії раціональних чисел, зводить питання про несуперечливість систем аксіом групи, кільця й поля до питання про несуперечливість теорії раціональних чисел; існування інтерпретації системи аксіом бульової алгебри в термінах теорії множин та логіки висловлень зводить питання про її несуперечливість до питання про

несуперечливість теорії множин чи логіки висловлень. В такому розумінні ці системи аксіом несуперечливі. Властивості повноти жодна з цих систем аксіом не має. Адже, як показують наведені вище приклади груп, кілець, полів і бульової алгебри, існують інтерпретації систем аксіом, що визначають ці структури, які складаються з різного скінченного числа елементів, а також інтерпретації, які складаються з нескінченної сукупності елементів. Такі інтерпретації не можуть бути ізоморфними між собою вже тому, що між множинами з різним скінченним числом елементів, а також між скінченною і нескінченною множинами не можна встановити взаємно однозначної відповідності.

Жодна з цих систем аксіом також не є незалежною. Проте на доведенні цього ми спинятися не будемо.

Алгебраїчні структури відіграють важливу роль в усіх галузях математики, але не вичерпують фундаментальних понять сучасної математичної науки. До цих понять відносяться також так звані структури порядку, топологічні структури та різноманітні структури, утворені внаслідок певного сполучення, комбінування властивостей кількох з названих структур. Деяке уявлення про структури порядку дають упорядковані множини, а про комбіновані структури – упорядковані кільця і поля, в яких сполучено властивості алгебраїчної структури і структури порядку.

Слід зауважити лише те, що це можна зробити на основі теорії множин, оскільки довільні відношення між елементами множини, внутрішні і зовнішні закони композиції для елементів множини можна означити за допомогою певних множин (областей задання та графіків відношень і відповідностей тощо). Тому математичну структуру можна задати системою множин, серед яких виділяється основна множина («носій структури»); відношення і внутрішні закони композиції задається графіками-підмножинами декартових степенів основної множини; для означення зовнішніх законів композиції задають допоміжні множини операторів та графіки-підмножини декартових добутків основної і допоміжних множин.

Поняття ізоморфізму також може бути узагальнене на випадок математичних структур довільного типу. Так, ізоморфізм структур порядку – це подібність відповідних упорядкованих множин; ізоморфізм упорядкованих алгебраїчних структур – це така взаємно однозначна відповідність між ними, що є одночасно ізоморфізмом щодо алгебраїчних операцій і подібністю (ізоморфізмом щодо відношення порядку). Спираючись на теоретико-множинне тлумачення відношень і операцій, можна звести поняття ізоморфізму двох структур до взаємно однозначної відповідності між основними і допоміжними множинами цих структур, при якій відповідні графіки відношень і функцій виявляються також поставленими у взаємно однозначну відповідність.

РОЗДІЛ 5.

ПОЛЕ РАЦІОНАЛЬНИХ ФУНКЦІЙ. ТИПИ ПОЛІВ. ОСНОВНІ ВЛАСТИВОСТІ ПОЛІВ.

5.1 Визначення поля.

Зупинимося детальніше на аксіомах, що визначають поняття поля.

Полем називається сукупність деяких елементів, для яких введено два закони композиції, що підкоряються наступним аксіомам:

Аксіома I. Всі елементи поля утворюють абелеву групу відносно першого закону композиції (означатимемо його знаком + і називати складанням).

Аксіома II. Усі елементи поля, за винятком одиниці першої групи (якою ми називатимемо нулем), утворюють абелеву групу відносно другого закону композиції (який ми називатимемо множенням).

Аксіома III. Добуток елементів поля дорівнює нулю тоді і тільки тоді, якщо один з них дорівнює нулю.

Аксіома IV. Має місце дистрибутивний закон:

$a\,(b + c) = ab + ac.$

5.1.1 Типи полів.

Сукупності 1) усіх дійсних чисел, 2) усіх комплексних чисел, 3) усіх чисел алгебри, тобто чисел, що задовольняють рівнянням алгебри з раціональними коефіцієнтами, є також полями.

Сукупності раціональних функцій від однієї або декількох змінних, коефіцієнти яких є раціональними числами (чи взагалі належать якому-небудь заданому полю), називається полем раціональних функцій.

Нехай дано одне або декілька рівнянь алгебри з раціональними коефіцієнтами. Сукупність раціональних функцій від коренів цих рівнянь складає поле, що називається полем алгебраїчних чисел, що і становитме предмет даного дослідження.

Сукупність раціональних функцій від декількох змінних, звязаних однією або декількома залежностями алгебри, називаєтся полем функцій алгебри.

Існують поля, що складаються з кінцевого числа елементів. Прикладом таких полів може служити сукупність класів порівнянь по простому модулю. Причина, чому сукупність класів порівнянь по складеному модулю не є полем, полягає в тому, що аксіома III вимагає, щоб добуток двох елементів поля дорівнював нулю тоді і тільки тоді, якщо принаймні один з множників дорівнює нулю. В той же час порівняння

$x*y=0(mod\ mn)$ може виконуватись, якщо ми покладемо $x = m$, $y = n$, причому ні m ні n не порівнянні з 0 при модулю mn.

Кінцеві поля найзагальнішого вигляду можна отримати при розгляді раціональних функцій від коренів поліномів, незвідних по простому модулю p, тобто незрівняних по модулю p з добутком поліномів нижчого степеня, і при цьому відкидати кратності p. Вони носять назву полів Галуа або полів порівнянь по подвійному модулю.

Гензель (K. Hensel) ввів у розгляд поля, природа яких абсолютно відмінна від природи числових і функціональних полів, так звані поля p-адичних чисел. Елементами поля p-адичних чисел є нескінченні ряди $a_0 + a_1p + a_2p^2 + ..$, розташовані по зростаючих степенях простого числа p з цілими раціональними коефіцієнтами. Такі ряди завжди розбігаються. Проте над цими рядами можна виконувати формальні дії додавання і множення, які задовольняють усім аксіомам поля, p-адичні числа служать хорошим інструментом при вивченні властивостей звичайних чисел алгебри.

5.1.2 Властивості полів чисел алгебри

Поля чисел алгебри розглядаються як сукупності раціональних функцій від одного або декількох коренів рівнянь алгебри. Можна показати, що будь-яке таке поле може бути утворене коренем одного рівняння. Іншими словами, справджується наступна теорема:

Теорему. У всякому полі, утвореному коренями $a_1, a_{2,...}a_m$ рівнянь алгебри $f_1(x)=0, f_2(x)=0,...,f_m(x)=0$ з раціональними коефіцієнтами (які можуть частково або повністю співпадати), існує примітивний елемент, тобто елемент, через який можуть бути раціонально виражені усі елементи поля.

Наслідок. Якщо $x, y, z,..$- величини, що утворюють поле, то примітивну величину поля можна підшукати у вигляді $c_1x + c_2y + c_3z + ..$, де $c_1\ c_2, c_3,..$ спеціально підібрані раціональні числа.

Теорема. Будь-яка величина поля чисел алгебри задовольняє рівнянню алгебри з раціональними коефіцієнтамии.

Степінь рівняння, що є незвідним і якому задовольняє примітивна величина поля, не залежить від вибору примітивної величини і носить назву степеня поля.

Теорема. Якщо $\theta_1,\theta_2,..,\theta_n$ - корені незвідного полінома $f(x)$, а $R\ (x)$ - раціональна функція (знаменник якої не ділиться на f(x)), то поліном:

$(t-R(\theta_1))(t-R(\theta_2))...(t-R(\theta_n))$ є степенем незвідного полінома.

Теорема. Якщо 0 - примітивний елемент поля степеня n, то всякий елемент цього поля можна представити у вигляді полінома степеня не вище $n-1$:

$$a_0 + a_1\theta + .. + a_{n-1}\,\theta^{n-1},$$

де коефіцієнти $a_1, a_2, .., a_{n-1}$ - раціональні числа. Це представлення єдине.

Поняття незвідності можна узагальнити на випадок, коли коефіцієнти поліномів не є раціональними числами, а належать до заданого поля. Поліном зводиться в полі K, якщо його можна розкласти на добуток поліномів нижчого степеня, коефіцієнти яких є величини поля K.

Приклад. Поліном x^4+1 є незвідним в раціональному полі, але в полі $K\left(\sqrt{2}\right)$ (тобто поля раціональних функцій від $\sqrt{2}$) зводиться. Дійсно,

$$X^4+1=(x^2+\sqrt{2}\,*x+1)(x^2-\sqrt{2}\,*x+1)$$

Всі властивості, що мають місце для поліномів, що не зводяться в раціональному полі, зберігають свою силу й для поліномів, що не зводяться в поле K.

Проте ті практичні методи, які ми запропонували для распознавания поліномів, що наводилися і не наводилися, тут вже не можуть бути проведені. Основна причині цього полягає в тому, що кожне ціле число (тут ми не будемо останавливаться на цьому понятті) алгебри може бути представлене у вигляді твору двох цілих чисел незліченним числом способів, через існування так званих одиниць алгебри.

5.2. Група Галуа. Співвідношень між коренями поліномів.

Нехай рівняння (без кратних коренів): $F(x)=x^n+a_1x^{n-1}+...+a_{n-1}x+a_n=0$ має корені $x_1, x_2..., x_n$. Між цими коренями матимуть місце співвідношення, що мають форму поліномів з коефіцієнтами, раціонально залежними від коефіцієнтів a_1, a_2, …, a_n, а також від заданих величин, які утворюють поле, що називається областю раціональності. Прикладом можуть служити співвідношення між елементарними, симетричними функціями від коренів та з коефіцієнтами:

$$\begin{cases} x_1 + x_2 + \ldots + x_n = -a_1, \\ x_1x_2 + x_1x_3 + \ldots + x_{n-1}x_n = a_2 \\ x_2x_3\ldots x_n + x_1x_3\ldots x_n + \ldots + x_1x_2\ldots x_{n-1} = (-1)^{n-1}a_{n-1} \\ x_1x_2\ldots x_{n-1}x_n = (-1)^n a_n \end{cases} \qquad (5.1)$$

Між коренями $x_1\ x_2,..,\ x_n$ можна виконувати будь-які перестановки, при цьому співвідношення (5.1) буде залишатися справедливими. Однак існують рівняння, між коренями яких дане співвідношення не виконується. Прикладом може служити рівняння: $x^4 + x^3 + x^2 + x + 1 = 0$, що має, як відомо, корінь 5-го степеня з одиниці: $x_1=\varepsilon,\ x_2=\varepsilon^2, x_3=\varepsilon^3, x_4=\varepsilon^4 (\varepsilon = e^{\frac{2\pi}{5}})$

Між цими коренями виконуються наступні співвідношення:

$$x_1x_4=1,\ x_1^2x_3=1, x_1^3x_2=1, x_1^4x_4=1 \qquad (5.2)$$

Не всяка підстановка між коренями буде зберігати ці співвідношеня в силі. Наприклад, підстановка (x_1,x_2) порушує відношення, переводячи його ліву частину в $x_2\ x_4$ тобто величину, рівну ε, а не одиниці. Перевіривши кожну з 24 підстановок симетричної групи 4-го степеня, можна переконатися, що співвідношень (3.10) не порушують такі заміни: $(x_1x_2x_4x_3)\ (x_1x_4)\ (x_2x_3)\ (x_1x_2x_4x_2)$.

Ці підстановки утворюють циклічну групу. Нехай в загальному вигляді між коренями рівняння мають місце співвідношення:

$$\varphi_i(x_1x_2,\ldots,x_n)=0\ (i=1.2\ldots.) \qquad (5.3)$$

Відносно кожної підстановки симетричної групи між $x_1\ x_2,\ \ldots,x_n\ Xa$ завжди можна сказати (принаймні теоретично), чи порушує вона хоч одне з цих співвідношень, або зберігає всі їх у силі (тобто переводить кожне співвідношення (5.3) в яке-небудь з цих же співвідношення). Справедлива така

Теорема. Сукупність усіх підстановок, що не порушують всіх можливих співвідношень між коренями алгебраїчного доповнення, утворює групу.

Група підстановок, що не порушує всіх можливих співвідношень між коренями рівняння $f\ (x) = 0$, носить назву групи Галуа або просто групи цього рівняння.

З усіх співвідношень (5.3) розглянемо спочатку співвідношення, в які входить тільки один з коренів нашого рівняння, наприклад x_1. Нехай $\Phi(x_1)$ буде одне з таких співвідношень і $f_1\ (x) = 0$ буде незвідне рівняння, якому задовільняє x_1. Тоді $\Phi(x) = 0$ і $f_1 = 0$ мають спільні корені, а тому $\Phi(x)$ має ділитися на $f_1(x)$. Таким чином $\Phi(x) = 0$ має місце для всіх коренів незвідного множника полінома $f(x)$. З іншого боку, $f_1(x_1) = 0$ само по собі є відношення, яке порушується від всіх підстановок. Якщо, зокрема, рівняння (5.1) незвідне, то співвідношення, в які входить один корінь, не накладають ніякого обмеження на його групу Галуа. Якщо ж рівняння (5.1) зводиться, то відразу видно, що його група Галуа інтранзитивна.

Таким чином, ліва частина всякого співвідношення $\Phi(x_1) = 0$, має тільки один корінь x_1, і повинна ділитися на незвідний поліном $f_1(x)$, якому задовольняє x_1. Це можна записати так: $\Phi(x) = 0 \pmod{f_1(x)}$, де x тепер треба вважати змінною величиною. Поліном $f_1(x)$ називається першим основним модулем.

Розглянемо співвідношення типу (5.3), в які входять два кореня, наприклад x_1 і x_2. ліву частину рівняння потрібно розкласти на незвідні множники в полі $K(x_1)$ (тобто у полі раціональних функцій від x_1) і позначити той незвідний поліном, якому задовольняє x_2 через $f_2(x_1, x)$. Нехай $\Phi(x_1\ x_2) = 0$ з будь яким співвідношенням між коренями x_1 і x_2. Розглянемо поліном $\Phi(x_1, x)$, який можна вважати поліномом від змінної x з коефіцієнтами з поля $K(x_1)$. Тоді поліном $\Phi(x_1,x)$ повинен ділитися на $f_2(x_1,x)$:

$$\Phi(X_1, X) = 0 \pmod{f_2(x_1\ x)}.$$

Якщо $\psi(x_1,x) = 0 \pmod{f_2(x_1\ x)}$, то $\psi(x_1\ x_2) = 0$, так як $f(x_1,x_2) = 0$.

Нехай степінь полінома $f_1(x)$ є n_1 і степінь полінома $f_2(x_1\ x)$ є n_2. Далі можна вважати, що в $f_2(x_1\ x)$ величина x_1 входить як поліном. Вважаючи ξ_1 ξ_2 змінними, потрібно розділити $\Phi(\xi_1,\xi_2)$ на $f_2(\xi_1,\xi_2)$. Це можна виконати, не вводячи дрібних функцій від ξ_1 (коефіцієнтами при ξ_2^{n2} в поліномі $f_2(\xi_1,\xi_2)$ служать одиниці. Тоді можна отримати:

$$\Phi(\xi_1,\xi_2) = f_2(\xi_1,\xi_2) * q_2(\xi_1,\xi_2) + r(\xi_1,\xi_2) \qquad (5.4)$$

де поліном $r(\xi_1,\xi_2)$ має степінь, не вищий $n-1$ щодо ξ_2. Тому всі коефіцієнти при ступенях ξ_2 полінома $r(\xi_1,\xi_2)$, маючи з незвідним поліномом $f(\xi_1)$ спільні корені, діляться на нього, і таким чином поліном $r(\xi_1,\xi_2)$ може бути представлений в формі $f(\xi_1) \cdot q_1(\xi_1,\xi_2)$, де $q_1(\xi_1,\xi_2)$ - деякий поліном, так що формула (5.4) набуває вигляду:

$$\Phi(\xi_1,\xi_2) = f_1(\xi_1) * q_1(\xi_1,\xi_2) + f_2(\xi_1,\xi_2) * q_1(\xi_1,\xi_2) \qquad (5.5)$$

де $q_1(\xi_1,\xi_2)$ і $q_2(\xi_1,\xi_2)$ - деякі поліноми від ξ_1 і ξ_2

Якщо замінити x_1 на будь-який інший корінь x_i полінома $f_1(x_1)$, то $f_2(x_1, \xi_2)$ є той поліном, що не зводиться в полі $K(x_1)$. Нехай

$$f(x_2) = (\xi_2 - x_1,) f_2(x_1, \xi_2) f_2^{(1)}(x_1, \xi_2) \ldots f_2^{(k)}(x_1, \xi_2).$$

Тому завжди можна в $f_2(x_1, x_2) = 0$ (а отже і в будь-якому співвідношенні між x_1 і x_2) замінити x_1 на будь-який корінь полінома $f_1(x)$, а

x_2 - на якийсь інший корінь, відмінний від x_1, без того, аби співвідношення порушилося.

Нехай $f_3(x_1, x_2, \xi_3)$ буде поліномом, що не зводиться в полі $K(x_1, x_2)$, і якому задовольняє $\xi_3 = x_3$, і хай його степінь буде n_3. Якщо розділити поліном $\Phi(x_1, x_2, \xi_3)$ на $f_3(x_1, x_2, \xi_3)$, то можна отримати:

$$\Phi(x_1, x_2, \xi_3) = f_3(x_1, x_2, \xi_3) \cdot q_3(x_1, x_2, \xi_3) + r(x_1, x_2, \xi_3). \qquad (5.6)$$

Залишок $r_3(x_1, x_2, \ldots_3)$, будучи не вище степеня n_3 - 1, має спільний з $f(x_1, x_2, \xi_3)$ корінь $\xi_3 = x_3$. Тому якщо розділити $\Phi(\xi_1, \xi_2, \xi_3)$ на $f(\xi_1, \xi_2, \xi_3)$ (при змінних ξ_1, ξ_2, ξ_3, але по змінній ξ_3), то залишок $r(\xi_1, \xi_2, \xi_3)$ при довільному ξ_3 перетворюватиметься на нуль, якщо покласти $\xi_1 = x_1$, $\xi_2 = x_2$, а тому рівність $r(\xi_1, \xi_2, \xi_3) = 0$ є співвідношенням між x_1 і x_2 з ξ_3 у вигляді параметра, і поліном $r(\xi_1, \xi_2, \xi_3)$ може бути представлений у формі (5.5), де ξ_3 може входити в поліноми q_1 і q_2 у вигляді параметра. Підставляючи в (5.6), можна отримати:

$$\Phi(\xi_1, \xi_2, \xi_3) = f_3(\xi_1, \xi_2, \xi_3)\, q_3(\xi_1, \xi_2, \xi_3) + + f_1(\xi_1) q_1(\xi_1, \xi_2, \xi_3) + + f_2(\xi_1, \xi_2)\, q_2(\xi_1, \xi_2, \xi_3) \qquad (5.7)$$

Поліном $f_3(x_1, x_2, \xi_3)$ є незвідним в полі $K(x_1, x_2)$ дільником полінома $f(\xi_3)$:

$$f(\xi_3) = f_3(x_1, x_2, \xi_3)\,(\xi_3 - x_2)\,(\xi_3 - x_1)\,F(x_1, x_2, \xi_3) \qquad (5.8)$$

Ця рівність при довільному ξ_3 є співвідношенням між x_1 і x_2, а тому в ньому можна замінити корені x_1, x_2 іншими коренями x_i, x_j, якщо при цьому не порушуються співвідношення типу $\Phi(x_1, x_2) = 0$. Тому поліном $f(x_i, x_j, \xi_3)$ має коренями (відносно ξ_3) деякі з коренів $x_1, x_2, \ldots, x_n$, причому відмінні від x_i і x_j. Дійсно, переводячи в співвідношенні (5.8) x_1 в x_i і x_2 в x_j, можна отримати:

$$f(\xi_2) = f_3(x_i, x_j, \xi_3)\,(\xi_3 - x_i)\,(\xi_3 - x_j)\,F(x_i, x_j, \xi_3).$$

Тут ліва, а тому і права, частина має коренями. $x_1, x_2, \ldots, x_n$, причому x_i і x_j - корені множників $\xi_3 - x_i$ і $\xi_3 - x_j$ правої частини.

Позначаючи через $f_k(x_1, x_2, \ldots, x_{k-1}, \xi_k)$, що є незвідним у полі $K(x_1, x_2, \ldots, x_{k-1})$, дільник полінома $f(\xi_k)$, що має коренем $\xi_k = x_k$ і продовжуючи міркування для значень $k = 4, 5, \ldots, n$, можна переконатися, що ліва частина будь-якого співвідношення $\Phi(x_1, x_2, \ldots, x_n) = 0$ може бути представлена в наступній формі:

$$\Phi(\xi_1, \xi_2,..., \xi_n) = f_1(\xi_1)q_1(\xi_1, \xi_2, ..., \xi_n) +$$
$$+ f_1(\xi_1, \xi_2,) q_2(\xi_1, \xi_2, ..., \xi_n) + ... + f_n(\xi_1, \xi_2, ..., \xi_n) q_n(\xi_1, \xi_2, ..., \xi_n), \quad (5.9)$$

де $q_i(\xi_1, \xi_2, ..., \xi_n)$ $(i = 1, 2,..., n)$ - поліноми з раціональними коефіцієнтами. Таким чином, кожна дана підстановка між коренями $x_1, x_2,..., x_n$ зберігає в силі будь-які співвідношення між $x_1, x_2,..., x_n$ тоді і лише тоді, якщо вона зберігає в силі співвідношення

$$f_1(x) = 0, \; f_2(x_1,x_2) = 0, \qquad f_3(x_1,x_2,x_3) = 0,..., \; f_n(x_1, x_2, ..., x_n) = 0. \quad (5.10)$$

Ліві частини співвідношень (5.10) носять назву основних модулей рівняння (5.10). Вони характеризують групу Галуа в тому сенсі, що кожна підстановка належить до групи Галуа тоді і лише тоді, якщо вона зберігає в силі всі співвідношення (5.10).

Крім того, продовжуючи міркування, можна переконатися, що в групі Галуа існуватиме підстановка, що переводить x_1 у будь-який корінь x_{a1} першого модуля $f_1(\xi_1)$. Дійсно, підставляючи в $f_2(\xi_1, \xi_2) = x_{a1}$ можна отримати поліном, що має коренем якісь з величин $x_1, x_2, ..., x_n$, відмінні від x_{a1}

5.3 Основні властивості групи Галуа.

Теорема. Для того, щоб група рівняння $a_0x^n + a_1x^{n-1} + ... + a_{n-1}x + a_n = 0$ була транзитивна, необхідно і достатньо, щоб рівняння було незвідним.

Цю теорему потрібно узагальнити на випадок кратної транзитивності.

Теорема. Для того, щоб група Галуа рівняння $a_0x^n + a_1x^{n-1} + ... + a_{n-1}x + a_n = 0$ була k раз транзитивна, необхідно і достатньо, щоб перші його k основних модулів збігалися з модулями Коші.

Зв'язок між порядком групи Галуа і степенями основних модулів може бути сформульований у вигляді

Теорема. Порядок групи Галуа дорівнює добутку $n_1.n_2...n_n$ степенів основних модулів.

Якщо група Галуа є симетрична, то вона n раз транзитивна, і тому всі n основних модулів через співпадають з модулями Коші. Степені цих модулів рівні, відповідно, n, $n-1$, $n-2$,..., 3, 2, 1, і ще раз можна прийти до того, що порядок симетричної групи рівний n.

Якщо група Галуа є знакозмінною, то вона $n-2$ рази траизитивна, а тому перші n–2 основних модуля повинні збігатися з модулями Коші. Степінь (n–1)-го модуля Коші дорівнює двом. Якщо він не зводиться, то матимем симетричну групу (n-й модуль у всіх випадках першого степеня).

Для випадку знакозміної групи залишається одна можливість, саме:*(n*–1)-й модуль повинен розкладатися на лінійні множники. Тому степені послідовних основних модулів рівняння із знакозмінною групою Галуа такі: *n*, *n* – 1, *n* – 2,.., 3, 1, 1.

Наступна теорема є однією з найважливіших в теорії Галуа відносно симетричних функцій:

Теорема. Функція від коренів $x_1, x_2,..., x_n$ рівняння $a_0x^n + a_1x^{n-1} + ... + a_{n-1}x + a_n = 0$, що не змінює своєї величини при всіх підстановках групи Галуа, раціонально виражається через коефіцієнти цього рівняння.

Приклад. Розглянемо рівняння:

$$f(x) = x^{p-1} + x^{p-2} + ... + x + 1 = 0. \qquad (5.11)$$

Воно є незвідним. З іншого боку, позначаючи через x_1 його корінь, можна виразити через нього інші корені так: $x_2=x_1^2$, $x_3=x_1^2$,.... $x_{p-1}= x^{p-1}$.

Його перший основний модуль є $f(\xi_1)$, тоді як всі модулі, починаючи з другого, є лінійними поліномами. Дійсно, в полі $k(x_1)$ поліном $f(\xi)$ розкладається на лінійні множники: $f(\xi) = (\xi - x_1)(\xi - x_1^2)\ldots(\xi - x^{p-1})$.

Тому основними модулями рівняння (5.11) є наступні поліноми:

$$f_1 = \xi_1^{p-1} + \xi_2^{p-2} + \ldots + \xi_1 + 1,\ f_2 = \xi_2 - \xi_1^2, \ldots,\ f_{p-1} = \xi_{p-1} - \xi_1^{p-1}.$$

Таким чином група Галуа рівняння (5.11) має порядок p- 1 і вона є циклічною. Позначаючи через *g* - первісний корінь порівняння $x^{p-1} - 1 = 0 (\mathrm{mod}\ p)$, через *T* - підстановку, що переводить x_1 в x_1^g, x_1^g в $x_1^{g^2}$,..., звернемо увагу на те, що T^k переводить x_1 в x^{g^k} (k = 1, 2,..., p-1). Але оскільки g^{p-1} є найменшим степенем *g,* порівнянним з одиницею по модулю *p,* то звідси слідує, що T^{p-1} є найменший степінь підстановки $T=1$. Тому всі підстановки 1, T, T^2,..., T^{p-2} різні і таким чином вичерпують всю групу Галуа, порядок якої рівний теж *p*- 1.

Якщо всі корені якого-небудь рівняння можуть бути раціонально виражені через один корінь, то рівняння називається нормальним. Не важко переконатися, що рівняння нормальне тоді і лише тоді, якщо всі його основні модулі, починаючи з другого, лінійні.

5.4. Підстановки групи Галуа.

Нехай, наприклад, задані корені x_1, x_2, x_3, x_4. Якщо корені x_1, x_2, x_3, x_4 піддати підстановці (x_1, x_3), то величина $z_1 = x_1x_2 + x_3x_4$, перейде у $z_2 = x_3x_2 + x_1x_4$. шляхом заміни кореня x_1 на корінь x_3 і кореня x_3 – на корінь x_1.

Нехай a-величина поля, і хай $\varphi_1\ (x_1, x_2,..., x_n)$ і $\varphi_2\ (x_1, x_2,..., x_n)$- два її різні вирази через корені. Тоді рівність $\varphi_1\ (x_1, x_2,..., x_n) + \varphi_2\ (x_1, x_2,..., x_n)$ можна розглядати як співвідношення між коренями рівняння. Якщо застосувати до цього співвідношення будь-яку підстановку групи Галуа, то це воно не повинно порушитися, а тому і в правій, і в лівій частині отримається однакова величина.

5.4.1 Автоморфізм нормального поля.

Щоб підійти до нового визначення групи Галуа, не залежному від вихідного рівняння, потрібно ввести поняття автоморфізму поля. Нехай кожен елемент поля переходить в якійсь інший, визначений елемент того ж поля. Автоморфізмом поля можна вважати відображення, яке переводить величини α, β нашого поля відповідно в α_1, β_1 так, що сума $\alpha + \beta$ переводиться в $\alpha_1 + \beta_1$ і добуток $\alpha\,\beta$ - в $\alpha_1\,\beta_1$.

З цього визначення витікають наступні властивості автоморфізму:

1°. Різниця переходить в різницю. Дійсно, нехай $\alpha\to\alpha_1$, $\beta\to\beta_1$, і нехай α-$\beta\to x$. Тоді $\alpha=(\alpha - \beta) + \beta$ переходить в $x\to\beta_1$, оскільки $\alpha\to\alpha_1$, то повинно мати місце $\alpha_1 = x+ \beta_1$, $x=\alpha_1- \beta_1$, тобто α-$\beta\to\alpha_1$-β_1.

2°. Ділене переходить в ділене. Доведення аналогічне.

3°. Нуль переходить в нуль, одиниця−в одиницю. Дійсно, $\alpha\to\alpha_1$ то $0 = \alpha$-$\alpha\to\alpha_1$-α_1=0, і $1=\frac{\alpha}{\alpha}\to\frac{\alpha_1}{\alpha_1}=1$.

4°. Всі раціоналъні числа залишаються на місцях.

5°. Якщо $\alpha\to\alpha_1$ і $f(x)$ - раціональна функція від x з раціональними коефіцієнтами, то $f(\alpha)\to f(\alpha_1)$.

6°. Корені рівнянь алгебри з раціональними коефіцієнтами переходять в корені тих же рівнянь. Насправді, якщо $\alpha\to\alpha_1$, то $f(\alpha)\to f(\alpha_1)$. Якщо при цьому $f(\alpha) = 0$, то в силу $0\to0$ повинно бути також $f(\alpha_1) = 0$.

7°.Якщо $x_1\to x_1'$, $x_2\to x_2'$, $x_n\to x_n'$ і притому $\Phi\ (x_1, x_2,..., x_n) = 0$, то і $\Phi(x_1', x_2', ..., x_n') = 0$.

Таким чином якщо в даному полі лежать всі корені якого-небудь рівняння, то автоморфізм поля виробляє над цими коренями підстановки його групи Галуа. Розширюючи підстановки групи Галуа, можна отримати з них автоморфізми поля $K\ (x_1, x_2,..., x_n)$.

5.4.2. Нормальні поля. Теорема Лагранжа.

Аби можна було говорити про групу Галуа як про групу автоморфізму поля, необхідно, аби кожна підстановка групи Галуа не

виводила величин даного поля з його границь. Тому поле, яке містить один корінь якого-небудь рівняння, повинно містити і всі інші його корені. Поле такого роду називається нормальним. Тоді групою Галуа нормального поля є сукупність всього його автоморфізму. Справедливі такі теореми.

Теорема. Примітивний елемент ξ нормального поля задовольняє незвідному рівнянню з раціональними коефіцієнтами, степінь якого дорівнює порядку групи поля.

Теорема. Нормальне поле містить елементи, що належать до кожної заданої підгрупи його групи Галуа.

Теорема Лагранжа. Якщо a належить до підгрупи групи Галуа нормального поля K, а β не змінюється від підстановок групи, то β раціонально виражається через α.

РОЗДІЛ 6.

ГРУПА ГАЛУА.

6.1. Початкові визначення групи Галуа.

6.1.1. Корені полінома. Розкладання полінома на лінійні множники.

У курсі алгебри доводяться наступні теореми:

Теорема. Будь-яке рівняння

$$a_0 x^n + a_1 x^{n-1} + ... + a_{n-1} x + a_n = 0 \qquad (6.1)$$

з дійсними або комплексними коефіцієнтами має принаймні один корінь, тобто таке значення змінної x, яке, будучи підставлено в рівняння (6.1), перетворює його ліву частину в нуль.

Теорема (Безу). Якщо рівняння $f(x)= 0$ має корінь число a, то поліном $f(x)$ ділиться на $x-a$.

Звідси випливає наступна властивість поліномів n-ної степені: вони не можуть мати більш як n різних коренів.

Якщо деякі з величин $a_1,a_2,..,\ a_n$ будуть рівними один одному, то говорять, що поліном $f(x)$ має кратні корені. Корінь a називається k-кратним, якщо $f(x)$ ділиться на $(x - a)^k$, але не ділиться на $(x-a)^{k+1}$. Для розпізнавання кратного кореня існує наступний критерій.

Теорема. Якщо a є для полінома $f(x)$ коренем k кратності, то для його похідної $f(x)$ він є коренем $(k-1)$ кратності.

Розглянемо найбільший спільний дільник $f_1(x)$ поліномів $f(x)$ і $f'(x)$. Він містить тільки такі множники $x - a$, які входять в $f(x)$, і притому у степені, вищому, ніж перший. Якщо $x - a$ входить в $f(x)$ в k-тій степені, то в $f_1(x)$ він увійде в $(k - 1)$ степені. Якщо $f(x)$ не має кратного кореня, то $f_1(x)$ не матиме множників, лінійних відносно x, і тому дорівнюватиме сталому числу.

Нехай $x-a,\ x-a',\ x-a'',\ldots$ будуть множники, що входять в $f(x)$ першого степеня; $x-\beta,\ x-\beta',x-\beta'',\ldots$ в другій степені, і т. д.; нарешті, $x-\omega,\ x-\omega',x-\omega'',\ldots$ множники, що входять в $f'(x)$ в самому вищому, k-тому степені $(k\le n)$. Введемо позначення:

$$X_1 = (x-a)(x-a')(x-a'')..., \; X_2 = (x-\beta)(x-\beta')(x-\beta'')...,...,$$
$$X_k = (x-\omega)(x-\omega')(x-\omega'')....$$

Тоді поліном *f(x)* можна представити так: $f(x) = a_0 X_1 \cdot {X_2}^2 \cdot {X_3}^3 ... X_k^k$ Поліном $f_1(x)$ виразиться в наступному виді: $f_1(x) = X_2 \cdot {X_3}^2 ... X_k^{k-1}$ Якщо ввести в розгляд $f_2(x)$, найбільший спільний дільник полінома $f_1(x)$ і його похідної $f'(x)$, то: $f_2(x) = X_3 ... X_k^{k-2}$.

Якщо продовжити цей процес до отримання полінома, не залежного від x, то можна отримати: $f_{k-1}(x) = X_k$, $f_k(x) = const$.

Уміючи знаходити поліноми $f_1(x)$, $f_2(x)$,.., $f_k(x)$, можна виділити в поліномі *f(x)* його множники X_1 X_2,..., X_k. Насправді, з отриманих співвідношень ми матимемо:

$$\frac{f(x)}{f_1(x)} = \varphi_1(x) = a_0 \cdot X_1 X_2 X_3 ... X_{k-1} X_k, \; \frac{f_1(x)}{f_2(x)} = \varphi_2(x) = X_2 X_3 ... X_{k-1} X_k, ...,$$

$$\frac{f_{k-2}(x)}{f_{k-1}(x)} = \varphi_{k-1}(x) = X_{k-1} X_k, \; f_{k-2}(x) = \varphi_k(x) = X_k$$

звідки

$$X_1 = \frac{\varphi_1(x)}{a_0 \varphi_2(x)}, \; X_2 = \frac{\varphi_2(x)}{\varphi_3(x)}, ..., X_{k-1} = \frac{\varphi_{k-1}(x)}{\varphi_k(x)}, X_k = \varphi_k(x)$$

6.1.2. Найбільший спільний дільник поліномів. Алгоритм Евкліда.

Залишається показати, що, знаючи *f(x)* і *f'(x)*, можна знайти найбільший спільний дільник $f_1(x)$ за допомогою раціональних операцій. Для знаходження найбільшого спільного дільника двох поліномів *f(x)* і *g(x)* існує метод, що носить назву алгоритму Евкліда, або алгоритм послідовного ділення. Він полягає в наступному. Нехай степінь полінома *f(x)* більший (чи, принаймні, не менше) степеня полінома *g(x)*. Розділимо *f(x)* на *g(x)* і визначимо таким образом ділене $q_1(x)$ і залишок $r_1(x)$. Потім розділимо *g(x)* на залишок $r_1(x)$, і нехай ділене буде $q_2(x)$, а залишок- $r_2(x)$. Розділимо $r_1(x)$ на $r_2(x)$, і т. д. При продовженні процесу степені поліномів $r_i(x)$ увесь час зменшуватимуться, так що ми врешті-решт можна дійти або до постійного залишку, або до того, що один з послідовних залишків $r_{m-1}(x)$ розділиться на наступний залишок $r_m(x)$. На цьому процес припиняється.

На основі відомого зв'язку між діленим, дільником, часткою і залишком матимемо:

$$\begin{aligned} f(x) &= g(x)\cdot q_1(x) + r_1(x) \\ g(x) &= r_1(x)\cdot q_2(x) + r_2(x) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ r_{m-2}(x) &= r_{m-1}(x)q_m(x) + r_m(x) \\ r_{m-1}(x) &= r_m(x)q_{m+1}(x) \end{aligned} \qquad (6.2)$$

Таким чином, найбільший спільний дільник двох поліномів може бути знайдений шляхом раціональних операцій. Звідси можна зробити висновок, що заданий поліном, який містить кратні корені, за допомогою раціональних операцій можна розкласти на множники, кожен з яких вже не міститиме кратних коренів.

Виразимо $r_m(x)$ на підставі передостанньої рівності (6.2) через $r_{m-1}(x)$ і $r_{m-2}(x)$: $r_m(x) = r_{m-2}(x) - q_m(x)\cdot r_{m-1}(x)$.

Якщо підставии в цю рівність вирази $r_{m-1}(x)$ через $r_{m-2}(x)$ і $r_{m-3}(x)$ на підставі попередньої рівності, то можна отримати однорідний лінійний вираз $r_m(x)$ через $r_{m-2}(x)$ і $r_{m-3}(x)$ з поліномами від x в якості коефіцієнтів:

$$r_m(x) = [1 + q_{m-1}(x)q_m(x)]r_{m-2}(x) - q_m(x)r_{m-3}(x).$$

Продовжуючи процес, можна отримати формулу

$$r_m(x) = u(x)\cdot f(x) + \upsilon(x)\cdot g(x), \qquad (6.3)$$

де $u(x)$ і $v(x)$ -поліноми від x.

Зокрема, якщо $f(x)$ і $g(x)$ - взаємно-прості, то:

$$u(x)\cdot f(x) + \upsilon(x)\cdot g(x) = 1 \qquad (6.4)$$

6.1.3. Подання коефіцієнтів полінома через його корені. Симетричні функції.

Прирівнюючи в тотожності

$$x^n + a_1x^{n-2} + \ldots + a_{n-1}x + a_n = (x-a_1)(x-a_2)\ldots(x-a_n)$$

(тут покладаємо $a_0 = 1$) коефіцієнти при різних степенях x, можна прийти до наступних формул:

$$\begin{aligned} a_1 &= -(a_1 + a_2 + \ldots + a_n), \\ a_2 &= (a_1a_2 + a_1a_3 + \ldots + a_{n-1}a_n.) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n-1} &= (-1)^{n-1}(a_1a_2\ldots a_{n-1} + a_1a_2\ldots a_{n-2}a_n + \ldots + a_1a_3\ldots a_n + a_2a_3\ldots a_n) \\ a_n &= (-1)^n a_1a_2\ldots a_n \end{aligned} \qquad (6.5)$$

Праві частини цих формул є цілими раціональними функціями від коренів $a_1, a_2, \ldots, a_n$, що мають при цьому наступну властивість: вони не змінюються, якщо самим довільним чином переставляти один з одним корені $a_1, a_2 \ldots, a_n$. Функції, що мають таку властивість, носять назву симетричних. Функції, представлені у формулах (6.5), носять, крім того, особливу назву елементарно-симетричних функцій.

Неважко переконатися, користуючись формулами (6.5), що всяка раціональна функція від коефіцієнтів $a_1, a_2 \ldots, a_n$ може бути виражена як симетрична функція від коренів $a_1, a_2 \ldots, a_n$. Дещо важче довести зворотну до неї теорему

Теорема. Всяка раціональна симетрична функція від коренів $a_1, a_2 \ldots, a_n$, може бути представлена як раціональна функція від коефіцієнтів $a_1, a_2 \ldots, a_n$. Якщо при цьому її коефіцієнти при степенях $a_1, a_2 \ldots, a_n$ цілі числа, то і через $a_1, a_2 \ldots, a_n$ вона виражається з цілими коефіцієнтами.

Приклад.

$$f(x) = x^3 + ax^3 + bx + c = (x-a)(x-\beta)(x-\gamma), V = a^3 + \beta^2 + \gamma^2.$$

Модулі Коші такі:

$$X = x^3 + ax^3 + bx + c, X_1 = x^2 + (a+a)x + (a^2 + aa + b), X_2 = x + (a + \beta + c)$$

Замінимо у виразі V букву γ на x і розділимо його на X_2. Отримаємо в залишку $a^2\beta^2+(a+\beta+a)^2$. Замінимо в цьому виразі β на x і розділимо його на X_2. Отримаємо в залишку: a^2-2b. Сюди a не входить, так що це і є шуканий вираз V через a, β, γ: $a^2 + \beta^2 + \gamma^2 = a^2 - 2b$.

6.1.4. Результант.

Задано два поліноми:

$$f(x) = (x - x_1)(x - x_2)....(x - x_n)$$
$$g(x) = (x - y_1)(x - y_2)....(x - y_m)$$

Виведемо умову того, щоб вони мали принаймні один загальний корінь. Для цього необхідно і достатньо, щоб вираз:

$$R(f, g) = f(y_1) f(y_2)...f(y_m) \tag{6.8}$$

перетворювався в нуль.

Вираз (6.8) носить назву результанта поліномів *f(x)* і *g(x)*. Будучи симетричною функцією від кореня полінома (6.7), результант може бути представлений, як ціла раціональна функція від коефіцієнтів цього полінома, у вираз якого входять також коефіцієнти полінома (6.6).

У виразі (6.8) коефіцієнти обох поліномів грають однакову роль. Щоб переконатися в цьому, перепишемо його так:

$$\begin{aligned} R(f, g) = {} & (y_1 - x_1)(y_1 - x_2)...(y_1 - x_n) \\ & (y_2 - x_1)(y_2 - x_2)....(y_2 - x_n) \\ & ..... \\ & (y_m - x_1)(y_m - x_2)...(y_m - x_n) \end{aligned}$$

Збираючи множники по стовпцях і користуючись формулою (6.7), можна отримати:

$$R(f, g) = (-1)^{mn} g(x_1) g(x_{21}) \ldots g(x_n), \tag{6.9}$$

звідки:

$$R(f, g) = (-1)^{mn} R(f, g) \tag{6.10}$$

Результанти знаходять застосування головним чином в теорії виключення невідомих з систем рівнянь.

6.1.5. Дискримінант.

Дискримінантом називається узятий з відомим знаком результант від полінома і його похідної :

$$R(f, f') = f'(x_1)f'(x_2)...f'(x_n) \tag{6.11}$$

Оскільки поліном має зі своєю похідною загальний корінь тоді і тільки тоді, якщо він має кратний корінь, то перетворення на нуль дискримінанта необхідно і достатньо для того, щоб поліном мав кратні корені.

Перетворимо вираз (6.11). Для цього звернемо увагу на те, що:

$$f'(x^i) = \lim \frac{f(x) - f(x_i)}{x - x_i} = \lim \frac{(x - x_1)(x - x_2)...(x - x_n)}{x - x_{n\setminus i}} =$$

$$(x_i - x_i)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)$$

Підставляючи цей вираз у формулу (6.11), можна отримати добуток всіх різниць між коренями $x_1, x_2, ..., x_n$, причому кожна різниця входить двічі, і при цьому з протилежними знаками. Таким чином можна отримати:

$$R(f, f') = (-1)^{\frac{n(n-1)}{2}} \cdot \prod_{i>j} (x_i - x_j)^2 \tag{6.12}$$

.

Вираз $\prod_{i>j} (x_i - x_j)$ залишається незмінним при парних і змінює знак при непарних підстановках. Його квадрат якраз і називатиметься дискримінантом і буде позначатися через *D(f),* так що: $R(f, f') = (-1)^{\frac{n(n-1)}{2}} \cdot D(f)$

Якщо розглянути вираз дискримінанта від добутку двох або декількох поліномів ($F(x) = f(x)\, g(x)$), то:

$$D(F) = \pm R(F, F') = \pm \prod_{i=1}^{n} \left[f(x_i) g(x_i) \right]' \cdot \prod_{i=1}^{n} \left[f(y_i) g(y_i) \right]' =$$

$$= \pm \prod_{i=1}^{n} \left[f(x_i) g'(x_i) + f(y_i)' g(y_i) \right] \cdot \prod_{i=1}^{n} \left[f(y_i) g'(y_i) + f'(y_i) g(y_i) \right]$$

Тоді в силу $f(x_i) = 0$ і $g(y_i) = 0$, можна отримати:

$$D(F) = \pm\prod_{i=1}^{n} g(x_i) \cdot \prod_{i=1}^{n} f'(x_i) \cdot \prod_{i=1}^{m} f(y_i) \prod_{i=1}^{m} g'(x_i) = \pm R(f, f') \cdot R(g, g') \cdot R(f, g) \cdot R(g.f) \quad , \qquad (6.13)$$

або

$$D(f \cdot g) = \pm D(f) D(g) R^2(f, g) \qquad (6.14)$$

У цій формулі кожен з дискримінантів і результантів правої частини є ціла раціональна функція з цілими коефіцієнтами від коефіцієнтів поліномів *f(x)* і *g(x)*. Звідси випливає, що кожен з множників правої частини є дільник числа *D(f – g)*.

Формула (6.14) допускає просте узагальнення на випадок багатьох поліномів. Вважаючи $F(x) = f_1(x) f_2(x).., f_k(x)$, можна отримати:

$$D(f_1, f_2, \ldots, f_k) = \pm\prod_{i=1}^{k} D(f_i) \cdot \prod_{i>j} R^2(f_i, f_j) \qquad (6.15)$$

6.2. Звідні та незвідні поліноми.

Розглянемо спочатку поліноми з раціональними коефіцієнтами. Поліном називається звідним, якщо його можна представити як добуток поліномів з раціональними коефіцієнтами, кожен з яких був би нижчої степені, ніж початковий. Якщо ж це неможливо, то поліном називається незвідним.

Для знаходження методів, що дозволяють за допомогою кінцевого числа дій розрізняти звідні та незвідні поліноми, вигідно перейти до поліномів з цілими коефіцієнтами. Цього можна легко досягнути шляхом множення поліномів на спільний знаменник усіх коефіцієнтів. Крім того, зручніше мати справу з поліномами, у яких коефіцієнт при старшому членові дорівнює одиниці. Щоб перейти від довільного цілочисельного полінома

$$A_0 + A_{1y} + \ldots + A_{n-1} y^{n-1} + A_n y^n$$

до цілочисельного полінома із одиничним старшим коефіцієнтом, треба зробити підстановку $y=\frac{x}{A_n}$, а потім помножити цей поліном на A_n^{n-1}. Отимається поліном типу:

$$a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + x^n \qquad (6.16)$$

Ця рівність показує, що кожне значення $\varphi(x_1)$ шуканого полінома $\varphi(x_1)$ є дільником числа $f(x_1)$. Але оскільки кожне ціле число має лише скінченне число множників, то надається скінченне число різних припущень відносно величини $\varphi(x_1)$. Комбінуючи ці припущення для усіх $i = 0,1,2,\ldots, n$ (зважаючи, що кожне передбачуване значення $\varphi(x_1)$ може мати або позитивний, або негативний знак, що ще збільшує число комбінацій), можна отримати деяке (практично досить велике) число гіпотез відносно значень $\varphi(x_0),\varphi(x_1),\ldots,\varphi(x_n)$. Кожну з цих гіпотез необхідно перевірити, тобто: 1) користуючись формулою Лагранжа, побудувати поліном $\varphi(x)$, що дає випробовувані значення для $\varphi(x_0),\varphi(x_1),\ldots,\varphi(x_n)$; 2) поліном $f(x)$ повинен ділитися на побудований нами поліном $\varphi(x)$.

Якщо виявиться, що ні при якій з гіпотез поліноми $\varphi(x)$ не задовольняють цим вимогам, то поліном $f(x)$ не зв одиться.

Описаний метод вимагає великих викладень, оскільки число випробувань, рівне $2^{n+1}*\omega_0 *\omega_1\ldots\omega_n$ де ω_i -число множників числа $f(x_i)$ (включаючи само $f(x_i)$ і одиницю), може виявитися на практиці дуже великим. Тому існує багато різноманітних методів, які дозволяють за тими або іншими ознаками відразу виключати деякі з гіпотез. Для прикладу можна навести метод, що полегшує перевірку гіпотез у тому випадку, якщо як $x_0, x_1, x_2, \ldots, x_n$ послідовні числа натурального ряду (або взагалі якщо вони складають арифметичну прогресію). В цьому випадку немає необхідності будувати для кожної гіпотези відповідний поліном $\varphi(x)$, а досить переконатися, що різниці u-го порядку від послідовності $\varphi(x_0), \varphi(x_1),\ldots,\varphi(x_n)$ мають постійне значення. Насправді, перші різниці полінома u -го степеня $\varphi(x)$ є значеннями полінома: $\Delta\varphi(x)=\varphi(x+1)-\varphi(x)$, степінь якого на одиницю менший. Другі різниці є значеннями полінома $(u - 2)$ -ого степеня, і т. д. Нарешті різниці u-го порядку є значеннями полінома u - ого степеня, тобто постійної величини.

Для знаходження раціональних дільників полінома можна запропонувати інший метод. Якщо поліном $f(x)$ має множник u-ого степеня з раціональними коефіцієнтами, то, позначивши його корені, які є також коренями полінома $f(x)$, через $x_1, x_2,\ldots, x_u$, потрібно звернути увагу на те, що усі елементарно-симетричні функції $x_1, x_2, .., x_u$ є раціональні числа. Нехай $\xi_1= \psi(x_1,x_2,\ldots, x_u)$- одна з таких елементарно-симетричних функцій.

Здійснюючи над величинами $x_1,x_2,\ldots,x_n$ будь-які перестановки симетричної групи порядку $n!$ (тобто замінюючи їх через $x_1, x_2,\ldots, x_u, x_{u+1},.., x_n$), можна отримати $s = C_n^u = \frac{n!}{(n-u)!u!}$ різних функцій $\xi_1, \xi 2, \ldots, \xi_n$. Коефіцієнти полінома $F(\xi)=(\xi - \xi_1)(\xi - \xi_2)\ldots(\xi - \xi_3)$ є симетричними функціями від $x_1, x_2,.., x_n$ і тому раціонально виражаються через коефіцієнти полінома $f(x)$. Знайшовши ці коефіцієнти, можна звести задачу до знаходження раціонального кореня полінома $F(\xi)$. Насправді, знаючи раціональні корені для кожного з рівнянь, які відповідають усім елементарно-симетричним функціям від кореня шуканого полінома, відразу отримується і самий поліном.

6.2.1 Критерій незвідності Ейзенштейна.

Використання загального методу для знаходження раціональних множників полінома на практиці є доволі складним. В зв'язку з цим було запропоновано декілька нових критеріїв, що дозволяють в деяких випадках показувати, що цей поліном є незвідним. Приведемо один з таких критеріїв, який належить Ейзенштейну (Eisenstein) :

Теорема. Якщо усі коефіцієнти полінома

$$f(x) = a_0 + a_1x + .. + a_{n-1}x^{n-1} + x^n$$

окрім останнього $a_n = 1$, діляться на яке-небудь просте число p, але при цьому перший коефіцієнт a_0 не ділиться на p^2, то поліном $f(x)$ незвідний.

Приклад. Розглянемо поліном

$$\frac{x^p - 1}{x - 1} = x^{p-1} + x^{p-2} + \ldots + x + 1 \tag{6.17}$$

де p яке-небудь просте число. Цей поліном має корінь так званий корінь з одиниці і називається поліномом ділення круга. Щоб довести його незвідність, зробимо підстановку $x = z + 1$. Отримаємо:

$$\frac{(z+1)^p - 1}{z} = z^{p-1} + C_p^1 z^{p-2} + \ldots + C_p^{p-2} z + C_p^{p-1} \tag{6.18}$$

Усі коефіцієнти $C_p^i = \frac{p(p-1)\ldots(p-i+1)}{1\cdot 2\cdot 3\ldots i}(i=1,2,...,p-1)$ діляться на p, причому останній коефіцієнт $C_p^{p-1}=p$ ділиться на p, але не на p^2. Через це поліном (6.18) не зводиться, звідки також слідує незвідність полінома (6.17). Доведемо незвідність загальнішого полінома:

$$\frac{x^{p^n}-1}{x^{p^{n-1}}-1} = x^{p^{n-1}(p-1)} + x^{p^{n-1}(p-2)} + \ldots + x^{p^{n-1}} + 1 \tag{6.19}$$

Роблячи знову підстановку $x = z + 1$, отримаємо:

$$\frac{z^{p^n-1} + C_p^1 n z^{p^n-2} + \ldots + C_{p^n}^{p^n-2} z + C_{p^n}^{p^n-1}}{z^{p^n-1} + C_{p^{n-1}}^1 z^{p^{n-1}-2} + \ldots + C_{p^n-1}^{p^n-1-2} z + C_{p^n-1}^{p^n-1-1}}$$

Поліноми, що стоять в чисельнику і знаменнику цього виразу, мають коефіцієнти, що діляться на p, окрім старших коефіцієнтів. В той же час, поліном-чисельник ділиться на поліном-знаменник і, як неважко побачити з самого процесу ділення, в поліномі-частці теж усі коефіцієнти, окрім старшого,

діляться на p:

$$C_{p^n}^{p^n-1} : C_{p^{n-1}}^{p^{n-1}-1} = p^n : p^{n-1} = p$$

Вільний член цього полінома ми отримаємо, якщо в нашій дробовій частині покладемо $z = 0$. Звідси слідує, що цей поліном задовольняє умовам критерію Ейзенштейна і тому є незвідним.

Теорема. Нехай $f(x)$ є незвідний поліном. Якщо відомо, що поліном $F(x)$ має з $f(x)$ хоча б один загальний корінь, то звідси випливає, що $F(x)$ ділиться на $f(x)$. При цьому вважаємо, що коефіцієнти обох поліномів є раціональні числа.

Наслідок. Існує один і тільки один звідний поліном з раціональними коефіцієнтами, якому задовольняє задане алгебраїчне число.

РОЗДІЛ 7

ПОЛЯ ГАЛУА

7.1. Скінченні комутативні поля (поля Галуа).

Серед простих полів характеристики p вже зустрічалися поля зі скінченною кількістю елементів. Такі скінченні поля називаються полями Галуа в честь їх першого дослідника Евариста Галуа.

Нехай Δ – поле Галуа і q– число його елементів.

Характеристика поля Δ не може бути рівною нулю, тому що інакше в Δ містилося б порожнє поле П з характеристикою нуль, яке складається з нескінченного числа елементів. Нехай p – характеристика даного скінченного поля. Порожнє поле П ізоморфне тоді до кільця класів лишків кільця цілих чисел по модулю p і тому містить p елементів.

Так як в Δ взагалі є лише скінченне число елементів, то в цьому полі існує найбільша система з лінійно незалежних над П елементів $\alpha_1, \ ..., \ \alpha_n$. Тоді n – ступінь розширення $(\Delta : \text{П})$ і кожен елемент з Δ набуває виду:

$$c_1\alpha_1 + ... + c_n\alpha_n, \tag{7.1}$$

де коефіцієнти c_1 з поля П однозначно визначені.

Для кожного коефіцієнта c_1 є p можливих значень; звідси слідує, що ми маємо точно p^n виразів виду (7.1). Оскільки ці вирази і дають елементи поля, про які йде мова, то отримуємо рівність $q = p^n$.

Отже, число елементів скінченного поля являється степенем характеристики p; показник цього степеня дорівнює степеню розширення $(\Delta : \text{П})$.

Будь-яка структура після відкидання нуля перетворюється в деяку мультиплікативну групу. У випадку поля Галуа ця група абелева має порядок $q-1$; отже $\alpha^{q-1} = 1$ для кожного $\alpha \neq 0$.

В такому випадку рівняння $\alpha^q - \alpha = 0$ має корінь $\alpha = 0$. Звідси, всі елементи поля являються коренями многочлена $x^q - x$. Якщо $\alpha_1, \ ..., \ \alpha_q$ – елемент поля, то $x^q - x$ ділиться на

$$\prod_{I}^{q}(x - \alpha_i).$$

В силу рівності степенів виходить що

$$x^q - x = \prod_I^q (x - \alpha_i).$$

Тому Δ складається з усіх коренів одного-єдиного многочлена $x^q - x$, що приєднуються до поля П. Цими умовами поле Δ визначається однозначно з точністю до ізоморфізму. Звідси, при заданих числах p і n всі поля з p^n елементів ізоморфні.

Тепер потрібно показати, що для кожного $n > 0$ та для кожного p дійсно існує поле з $q = p^n$ елементів.

При розгляді простого поля П характеристики p можна побудувати над П поле, в якому многочлен $x^q - x$ повністю розкладається на лінійні множники. В цьому полі розглядається множина коренів многочлена $x^q - x$. Останнє являється полем тому, що з $x^{p^n} = x$ i $y^{p^n} = y$ випливає, що $(x - y)^{p^n} = x^{p^n} - y^{p^n}$, а у випадку $y \neq 0$ $\left(\frac{x}{y}\right)^{p^n} = \frac{x^{p^n}}{y^{p^n}}$.

Так що різниця і відношення двох коренів розглянутого многочлена знову являються його коренями.

Многочлен $x^q - x$ має тільки прості корені тому, що похідна, зважаючи на порівняння $q \equiv 0(p)$, дорівнює $qx^{q-1} - 1 = -1$, a -1 не є нуль. Отже, множина коренів являється множиною елементі поля з q елементів.

Отже, для кожного степеня простого числа $q = p^n (n > 0)$ існує одне, і з точністю до ізоморфізму тільки одне поле Галуа з q елементів. Ці елементи являються коренями многочлена $x^q - x$.

Поле Галуа з p^n елементів далі буде позначатися через $GF(p^n)$.

Нехай $q - 1 = h$ та відмітимо, що всі відмінні від нуля елементи поля Галуа являються коренями многочлена $x^h - 1$, тобто коренями h-ої степені з одиниці. Так як h i p взаємно прості, то для цих коренів з одиниці справджуються вищенаведені міркування.

Всі відмінні від нуля елементи поля являються степенями деякого примітивного поля h-го степені з одиниці. Це означає, що мультиплікативна група поля Галуа циклічна.

Якщо ζ – примітивний корінь h-ої степені з одиниці в $\Delta = GF(p^n)$, то всі ненульові елементи з Δ являються степенями елемента ζ. Звідси слідує, що Δ=П(ζ) і Δ являється простим розширенням поля П. Степінь елемента ζ над П рівна, зазвичай, степеню розширення n.

Цією теоремою будова скінченних полів описується повністю.

В подальшому знадобиться наступна

Теорема. Поле Галуа з характеристикою p містить разом з кожним своїм елементом a рівно один корінь p-го степеня з a.

Теореми справедливі для полів $GF(p^n)$, в частковому випадку $n=1$ стають теоремами про кільце класів лишків $Z/(p)$ і співпадають з теоремами, відомими з елементарної теорії чисел. А саме:

1. Порівняння по модулю p має найбільше число коренів по модулю p таке, яка його степінь.

2. Теорема Ферма:

$$a^{p-1} \equiv 1 (\mathrm{mod}\, p) \text{ для } a \neq 0 (\mathrm{mod}\, p).$$

3. Існує «первісний корінь ζ по модулю p» – таке число, що будь-яке число b, взаємно просте з p, порівняне за модулем p з деяким степенем числа ζ. (Інакше кажучи: група класів лишків за модулем p, відмінних від нуля, являється циклічною.)

4. Добуток всіх відмінних від нуля елементів $a_1 \quad a_2 \quad ... \quad a_h$ поля $GF(p^n)$ рівне -1, так як

$$x^h - 1 = \prod_I^h (x - \alpha_v).$$

Для $n=1$ це дає теорему Вільсона: $(p-1)! \equiv -1 (\mathrm{mod}\, p)$.

7.2. Сепарабельні і несепарабельні розширення.

Нехай Δ – поле. Потрібно вияснити, чи може незвідний в $\Delta[x]$ многочлен мати кратні корені?

Для того, щоб $f(x)$ мав кратні корені, многочлени $f(x) \quad i \quad f'(x)$ повинні мати відмінний від константи множник, який можна вирахувати вже в $\Delta[x]$. Якщо многочлен $f(x)$ не можна розкласти, то ні з яким многочленом меншого степеня $f(x)$ він не може мати несталих спільних множників, звідки має місце рівність $f'(x) = 0$.

Нехай:

$$f(x) = \sum_0^n a_v x^v$$

$$f'(x) = \sum_1^n v a_v x^{v-1}$$

Так як $f'(x) = 0$, в нуль може перетворюватись кожний коефіцієнт:

$$va_v = 0 \quad (v = 1, \quad 2, \quad ..., \quad n).$$

У випадку характеристики нулю звідси слідує, що $a_v = 0$ для всіх $v \neq 0$. Звідси, непостійний многочлен не може мати кратних коренів. У випадку ж характеристики p рівності $va_v = 0$, і для $a_v \neq 0$, але тоді обов'язково повинно виконуватися порівняння

$$v \equiv 0(p).$$

Таким чином, щоб многочлен $f(x)$ володів кратними коренями всі його доданки повинні перетворюватись в нуль, за виключенням тих $a_v x^v$, для яких $v \equiv 0(p)$, тобто $f(x)$ повинен мати вигляд

$$f(x) = a_0 + a_p x^p + a_{2p} x^{2p} + \ldots$$

Зворотно: якщо $f(x)$ має такий вигляд, то $f'(x) = 0$. В такому випадку ми можемо записати:

$$f(x) = \varphi(x^p).$$

Тим самим доведено твердження: у випадку характеристик нуль незвідний в $\Delta[x]$ многочлен $f(x)$ має тільки прості корені; у випадку характеристики p многочлен $f(x)$ (якщо він відмінний від константи) має кратні корені тоді і тільки тоді, коли його можна представити як многочлен φ від x^p.

В останньому випадку, може здатися, що $\varphi(x)$ в свою чергу являється многочленом від x^p. Тоді $f(x)$ являється многочленом від x^{p^2}. Нехай $f(x) = \psi(x^{p^e})$, але не являється многочленом від $x^{p^{e+1}}$. Звісно, многочлен $\psi(y)$ незвідний. Далі, $\psi'(y) \neq 0$, тому, що інакше $\psi(y)$ мав би вигляд $x(y^p)$, і звідси, $f(x)$ представлявся б у вигляді $\chi(x^{p^{e+1}})$, що протирічить припущенню. Звідси $\psi(y)$ має лише прості корені.

Розкладемо многочлен $\psi(y)$ в деякому розширенні основного поля на лінійні множники:

$$\psi(y) = \prod_1^m (y - \beta_i),$$

тоді:

$$f(x) = \prod_1^m \left(x^{p^e} - \beta_i\right).$$

Нехай α_i – деякий корінь многочлена $x^{p^e} - \beta_i$. Тоді

$$\alpha_i^{p^e} = \beta_i$$

$$x^{p^e} - \beta_i = x^{p^e} - \alpha_i^{p^e} = (x - \alpha)^{p^e}.$$

Звідси, α_i являється p^e - кратним коренем многочлена $x^{p^e} - \beta_i$ і

$$f(x) = \prod_1^m (x - \alpha_i)^{p^e} .$$

Всі корені, многочлена $f(x)$ мають, таким чином, одну і ту ж кратність p^e.

Степінь m многочлена ψ називається редукованим степенем многочлена $f(x)$(або кореня α_1); число e називається показником многочлена $f(x)$(або кореня α_1) над полем Δ. Між степенем, редукованим степенем та показником має місце співвідношення $n = mp^e$, де m рівне числу різних коренів многочлена $f(x)$.

Якщо θ – корінь незвідного в кільці $\Delta[x]$ многочлена, що має лише прості корені, то θ називається сепарабельним елементом над Δ або елементом першого роду над Δ. При цьому незвідний многочлен, всі корені котрого сепарабельні, називається сепарабельним. В протилежному випадку алгебраїчний елемент θі незвідний многочлен $f(x)$ називаються несепарабельними або елементом (відповідно, многочленом) другого роду. Врешті, алгебраїчне розширення $\sum$, всі елементи якого сепарабельні над Δ, називається сепарабельним на Δ, а будь-яке інше алгебраїчне розширення називається несепарабельним.

У випадку характеристики нуль, згідно із сказаним вище, кожний незвідний многочлен (а тому і кожне алгебраїчне розширення), являється сепарабельним. Пізніше буде показано, що більшість найважливіших і цікавих розширень полів сепарабельні, а також існування цілих класів полів таких, які взагалі не мають несепарабельних розширень (так звані «досконалі поля»).

Розглянемо тепер алгебраїчне розширення $\sum = \Delta(\theta)$. Коли степінь n рівняння $f(x) = 0$, що визначає це розширення, дорівнює степеню $(\sum : \Delta)$, то редукована степінь m виявляється рівною числу ізоморфізмів поля $\sum$ в такому змісті: розглянемо такі ізоморфізми $\sum \cong \sum'$, при яких елементи підполя Δ залишаються нерухомими, і відповідно, $\sum$ переводиться в еквівалентне поле $\sum$ (ізоморфізм поля $\sum$ над полем Δ) та при яких поле-образ $\sum$ лежить всередині деякого спільного для них поля Ω. В таких випадках має місце твердження: при підходящому виборі поля Ω

розширення $\sum = \Delta(\theta)$ має рівно *m* ізоморфізмів над Δ та при будь-якому виборі поля Ω поле $\sum$ не може мати більше *m* таких ізоморфізмів.

7.3. Досконалі і недосконалі поля.

Поле Δ називається досконалим, якщо будь-який незвідний в $\Delta[x]$ многочлен $f(x)$ сепарабельний. Всі інші поля називаються недосконалими.

Умови, при яких поле являється досконалим, описуються таким чином.

Теорема. Поле характеристики нуль завжди досконале.

Доведення наведено в попередньому пункті.

Теорема. Поле характеристики *p* являється досконалим тоді і тільки тоді, коли воно разом з кожним своїм елементом місить і корінь *p*-го ступеня з нього.

Доведення. Якщо разом з кожним елементом поля мати корінь *p*-го степеня з нього, то кожний многочлен $f(x)$, який містить лише степені елемента x^p, являється *p*-им степенем, так як

$$f(x) = \sum_k a_k (x^p)^k = \sum_k \left\{\sqrt[p]{a_k}\, x^k\right\}^p = \sum_k \left\{\sqrt[p]{a_k}\, x^k\right\}^p .$$

Тобто кожний незвідний многочлен являється в такому випадку сепарабельним.

З іншого боку, якщо в полі є елемент α, корінь *p*-го степеня, з якого в полі не міститься, то розглянемо многочлен $f(x) = x^p - \alpha$.

Нехай $\varphi(x)$ – незвідний дільник многочленна $f(x)$. Після приєднання елемента $\sqrt[p]{\alpha} = \beta$ многочлен $f(x)$ розкладається на рівні лінійні множники $(x-\beta)$, тобто $\varphi(x)$, будучи дільником $f(x)$, являє собою деякий степінь двочлена $(x-\beta)$. Якби $\varphi(x)$ був лінійним, тобто $\varphi(x) = x - \beta$, то елемент β належав би полю Δ, що суперечить умові. Звідси, $\varphi(x) = (x-\beta)^k$ при $k > 1$ – деякий несепарабельний многочлен над Δ, а тому Δ – недосконале поле. Між іншим, степінь многочлена $\varphi(x)$ обов'язково ділиться на *p*, а тому в цьому випадку вона просто рівна *p*, тобто $\varphi(x) = f(x)$.

З наведених вище теорем слідує висновок, що всі поля Галуа досконалі.

Поле Ω називається алгебраїчно замкнутим, якщо кожний многочлен з кільця $\Omega[x]$ розкладається на лінійні множники. В кожному такому полі будь-який незвідний многочлен лінійний. Отже, всі алгебраїчно замкнуті поля – досконалі.

З визначення досконалого поля одразу отримаємо дві теореми.

Теорема. Кожне алгебраїчне розширення досконалого поля сепарабельне над цим полем.

Теорема. Для будь-якого недосконалого поля існує несепарабельне розширення.

Справді, ці несепарабельні розширення одержуються додаванням кореня будь-якого незвідного несепарабельного многочлена.

В досконалому полі характеристики *p* кожний многочлен $f(x)$, який залежить лише від x^p, являється *p*-им степенем. Дане твердження зберігає силу і для випадку многочлена від декількох змінних $f(x,y,z,...)$, що являється в дійсності многочленом від x^p, y^p, z^p, ... Для полів характеристики *p* ця властивість використовується часто.

7.4. Простота алгебраїчних розширень. Теорема про примітивний елемент.

Вияснимо тепер, в яких випадках розширення Σ поля Δ буде простим, тобто отримується приєднанням примітивного елемента. З цього слідує наступна теорема про примітивний елемент, яка справедлива для досить широкого класу випадків.

Теорема. Нехай $\Delta(\alpha_1,...\alpha_n)$ - кінцеве алгебраїчне розширення поля Δ і $\alpha_2,...,\alpha_n$ - сепарабельні елементи. Тоді $\Delta(\alpha_1,...,\alpha_n)$ є простим розширенням:

$\Delta(a_1,...,a_n)=\Delta(\theta)$.

Доведення. Доведення доцільно привести спочатку для двох елементів α,β, з яких β - сепарабельний. Нехай $f(x)=0$ - рівняння яке не розкладається для елемента α і $g(x)=0$ - рівняння яке не розкладається для елемента β. Перейдемо до поля, в якому $f(x)$ і $g(x)$ розкладаються. Нехай $a_1,...,a_r$ - різні корені многочленна $f(x)$, а $\beta_1,...,\beta_s$ - корені многочлена $g(x)$. Нехай $\alpha_1=\alpha,\beta_1=\beta$.

Припускається, що поле $\Delta(\alpha,\beta)$ також могло бути кінцевим, а для кінцевих полів існування примітивного елемента (навіть примітивного кореня з одиниці, степенями якого є всі ненульові елементи поля) вже було доведено.

Для $k\neq 1$ має місце нерівність $\beta k\neq\beta_1$, тому рівняння

$$\alpha_i+x\beta_k=\alpha_1+x\beta_1$$

для будь-якого i кожному $k\neq 1$ має щонайбільше один корінь x в Δ. Вибирається елемент c з відмінним від всіх коренів цих лінійних рівнянь; тоді для всіх i і $k\neq 1$, а також $\alpha 1+c\beta k\neq\alpha 1+c\beta 1$.

Нехай $\theta = \alpha_1 + c\beta_1 = \alpha + c\beta$, тоді θ є елементом поля $\Delta(\alpha, \beta)$. Слід показати, що θ має властивості шуканого примітивного елемента: $\Delta(\alpha, \beta) = \Delta(\theta)$. Елемент β задовольняє рівняння:

$$g(\beta) = 0,$$
$$f(\theta - c\beta) = f(\alpha) = 0.$$

Коефіцієнт яких лежить в $\Delta(\theta)$. Многочлени $g(x), f(\theta - cx)$ мають загальний лише корінь β, тому що для інших коренів $\beta k (k \neq 1)$ першого рівняння маємо:

$$\theta - c\beta_k \neq \alpha_i \quad (i = 1,..., r)$$

і відповідно,

$$f(\theta - c\beta_k \neq 0).$$

Елемент β є простим коренем многочленна $g(x)$; відповідно $g(x)$ і $f(\theta - cx)$ мають спільним лише один лінійний множник $x - \beta$. Коефіцієнти цього найбільшого спільного дільника повинні лежати в $\Delta(\theta)$; відповідно β лежить в $\Delta(\theta)$.

Із рівняння $\alpha = \theta - c\beta$ те саме випливає для α, тобто, $\Delta(\alpha, \beta) = \Delta(\theta)$.

Тим самим теорема доведена для $h = 2$. Якщо рахувати її доведеною для $h - 1$, то $\Delta(\alpha_1,..., \alpha_h - 1) = \Delta(\eta)$

і відповідно,

$$\Delta(\alpha_1,..., \alpha_h) = \Delta(\eta, a_h) = \Delta(\theta).$$

В відповідності з вже доведеною частиною теореми, теорема справджується і для h.

Наслідок. Кожне кінцеве сепарабельне розширення є простим.

Цей наслідок спрощує вивчення кінцевих сепарабельних розширень, тому що будова і ізоморфізми цих розширень дуже легко описуються через представлення базисів

$$\sum_0^{n-1} a_k \theta^k.$$

Наприклад, доведення твердження здійснюється з допомогою послідовного продовження ізоморфізмів: кінцеве сепарабельне розширення $\sum$ поля Δ має стільки ж ізоморфізмів над Δ, яка степінь $(\sum : \Delta)$. Дійсно, для простих сепарабельних розширень це твердження вже було доведено тобто, всяке кінцеве сепарабельне розширення є простим.

7.5. Норми і сліди.

Нехай $\sum$ - кінцеве розширення поля Δ або деяке кільце, яке є одночасно кінцевомірним векторним простором над Δ. Тоді елементи кільця $\sum$ можуть бути вираженими через n базисних елементів $u_1,...,u_n$ з коефіцієнтами із Δ $u = u_1c_1 + ... + u_nc_n$.

Для довільних t,u,υ із $\sum$ які задовольняють співвідношенням:

$$t(u+\upsilon) = tu + t\upsilon$$

$$t(uc) = (tu)c \qquad (c \in \Delta).$$

Таким чином, множення з ліва на t є лінійним перетворенням простору $\sum$ в себе. Матриця T цього лінійного перетворення в базисі $u_1,...,u_n$ визначається умовою $tu_k = \sum u_i t_{ik}$.

Визначник $D(T)$ цієї матриці, не залежить від вибору елементу базису, називається регулярною нормою або просто нормою елемента t в розширенні Σ поля Δ:

$$N(t) = D(T) = Det\left\|t_{ik}\right\|. \tag{7.2}$$

Норму можна визначити як визначник векторів tu_k відносно базису $u_1 \quad ... \quad u_n$:

$$N(t) = D(tu_1 \quad ... \quad tu_n). \tag{7.3}$$

Слід $S(T)$ матриці Т також не залежить від вибору базису; цей елемент основного поля називається регулярним слідом або просто слідом елементу t розширення Σ над полем Δ:

$$S(t) = S(T) = \sum t_{kk}. \tag{7.4}$$

Якщо елементу t відповідає матриця Т, то елементу t' – матриця Т', по добутку tt' відповідає матриця ТТ', а сумі $t + t'$ – сума Т+Т'. Звідси:

$$N(tt') = N(t)N(t'), \tag{7.5}$$

$$S(t + t') = S(t) + S(t'). \tag{7.6}$$

Починаючи звідси Σ є деяким тілом, в центрі якого міститься поле Δ, тобто завжди $cu = uc$ для $c \in \Delta, \quad u \in \Sigma$.

Кожний елемент t з Σ міститься в деякому комутативному тілі $\Delta(t)$ та існує мінімальний многочлен $\varphi(z) = z^m + a_1 z^{m-1} + ... + a_m$ з властивістю $\varphi(t) = 0$. Будова простого розширення $\Delta(t)$ повністю визначається мінімальним многочленом, і звідси, норму і слід елементу t в розширенні $\Delta(t)$ можна вирахувати через коефіцієнти мінімального многочлена.

В якості базису $u_1, \ ..., \ u_n$ розширення $\Delta(t)$ вибирається набір:

$$1, \ t, \ t^2, \ ..., \ t^{m-1}. \tag{7.7}$$

Якщо базисні вектори помножити помножити на t, то отримаємо набір:

$$t, \ t^2, \ t^3, \ ..., \ t^m. \tag{7.8}$$

Тепер, можна виразити вектори через базисні вектори, тоді:

$$\begin{aligned}
t &= \quad t, \\
t^2 &= \quad t^2, \\
&. \ . \ . \ . \ . \ . \ . \ . \\
t^{m-1} &= \quad t^{m-1}, \\
t^m &= -a_m 1 - a_{m-1} t - a_{m-2} t^2 - ... - a_1 t^{m-1}.
\end{aligned}$$

Сума діагональних елементів матриці перетворення дорівнює $-a_1$; отже слід елемента t розширення $\Delta(t)$ рівний:

$$s(t) = -a_1. \tag{7.9}$$

Норма елемента t в розширенні $\Delta(t)$ є визначником векторів (7.8) $n(t) = D(t, t^2, ...t^m)$.

Якщо змінити цей визначник відповідно з правилами дій над визначниками, то:

$$n(t) = (-1)^{m-1} D(t^m, t, t^2, ...t^{m-1}). \tag{7.10}$$

Після цього виразивши t^m через 1, t … t^{m-1}:

$$t^m = -a_m 1 - a_{m-1} t - a_{m-2} t_{m-2} t^2 - ... - a_1 t^{m-1}. \tag{7.11}$$

Визначник з двома однаковими стовбцями рівний нулю, через це із всіх доданків в правій частині приймається до уваги лише перше. Тоді отримуємо рівність:

$$\begin{aligned}
n(t) &= (-1)^{m-1} D(-a_m, 1, t^2, ..., t^{m-1}) = \\
&= (-1)^m a_m D(1, t, t^2, ..., t^{m-1}),
\end{aligned}$$

або так як визначник з базисних векторів рівний одиниці,

$$n(t) = (-1)^m a_m. \tag{7.12}$$

Слід і норма елемента t в полі $\Delta(t)$ є, таким чином, з точністю до знака другим і останнім коефіцієнтами в мінімальному многочлені $\varphi(z)$.

В деякому підходящому образі вибраному розширенні поля $\Delta(t)$ мінімальний многочлен $\varphi(z)$ розкладається на лінійні множники:

$$\varphi(z) = (z - t_1)...(z - t_m) \quad (t_1 = t). \tag{7.13}$$

Тоді

$$n(t) = (-1)^m a_m = t_1 t_2 ... t_m, \tag{7.14}$$

$$s(t) = -a_1 = t_1 + t_2 + ... + t_m. \tag{7.15}$$

Отже, норма і слід елемента t в розширенні $\Delta(t)$ над Δ виявляються рівними добутками і сумі елементів $t_1,...,t_m$, сполучених з t в полі розкладу многочлена $\varphi(z)$, при чому кожен сполучений елемент t_i береться стільки разів, скільки раз відповідний множник з t_i входить в розкладення. Якщо елемент t сепарабельний над Δ, то кожен зв'язаний елемент береться один раз.

Цим методом, але тільки з деякими більш громіздкими розрахунками, отримується норма $N(t)$ і слід $S(t)$ елемента t в розширенні Σ. Якщо знову $m-$ степінь розширення $\Delta(t)$ над Δ і g - степінь розширення Σ над $\Delta(t)$, то $n = mg$ - степінь розширення Σ над Δ. Базис розширення $\Delta(t)$ поля Δ складають степені. Нехай $v_1,...,v_g$ - деякий базис розширення Σ поля $\Delta(t)$. Тоді добуток $1v_1, tv_1,...,t^{m-1}v_1;1v_2,...;1v_g,...,t^{m-1}v_g$, складають деякий базис поля Σ над полем Δ. Якщо помножити базисні елементи з ліва на t і виразити добуток знову через цей базис, то сума діагональних елементів виявиться рівною

$$S(t) = (-a_1) + ... + (-a_1) = g(-a_1)$$

або

$$S(t) = gs(t). \tag{7.16}$$

Визначник базисних елементів, перемножених на t, рівне:

$$N(t) = D(tv_1, t^2 v_1,...,t^m v_1;...;tv_g,...,t^m v_g) =$$

$$= (-1)^{g(m-1)} D(t^m v_1, tv_1, t^2 v_1,...;...;t^m v_g, tv_g,...,t^{t-1} v_g).$$

Виразивши t^m через 1, t, ... t^{m-1} і скориставшись теоремами про визначники, тоді:

$$N(t) = (-1)^{gm} a_m^g = \left\{(-1)^m a_m\right\}^g,$$

або

$$N(t) = n(t)^g. \tag{7.17}$$

Отже, норма в розширенні Σ є g-им степенем норми в розширенні, а слід є g-кратним сліду в $\Delta(t)$.

РОЗДІЛ 8

ТЕОРЕТИЧНІ ЗАСАДИ, ПРИНЦИПИ ПОБУДОВИ ТА КЛАСИФІКАЦІЯ КОДІВ ПОЛЯ ГАЛУА

8.1. Базис Уолша – теоретична основа кодів поля Галуа.

Отже, теоретичною основою кодів поля Галуа є теорія чисел, теорія структур, груп та полів Галуа. Одночасно теоретичні засади кодів поля Галуа базуються на системі ортогональних дискретних функцій базису Уолша, у середовищі яких існує система квазістаціонарних рекурентних функцій Галуа.

Система функцій Уолша є повною системою ортонормованих прямокутних функцій. Існують різні впорядкування такого класу базисних функцій по Уолшу, по Петлі та інш.

При впорядкуванні по частоті, яке вперше дослідив Р. Уолш множина функцій, впорядкованих таким чином, позначається:

$$Sw = \{wal_w(i,t); i \epsilon \epsilon, \overline{N-1},$$

де: $N=2^n$; n=1.2…; w-ознака впорядкування по Уолшу.

Частота S_i функцій wal_w (i,t) визначається умовами:

$$Si = \begin{cases} 0, i = 1; \\ i/2, i - парне; \\ (i+1)/2, i - непарне. \end{cases}$$

Функції *cal*(S_i, t) і *sal*(S_i, t), які відповідають Wal_w(i, t) описуються наступним чином:

cal(S_i,t) = wal (i,t), i-парне;
sal(S_i,t) = wal (i,t), i- непарне.

Базис Уолша при відповідному впорядкуванні формує базиси Адамара і Галуа. При цьому породжуються коди Грея, Хемінга, М-послідовності, коди Баркера і коди поля Галуа.

Базис Галуа заданий системою дискретно-постійних функцій *sal*(z,n,t) та інтервалом обмежень аргументу [0, 2^{n-1}] та [0, 2^n], які визначені для будь-яких натуральних чисел w і n при $0 \le w \le 2^n$ згідно наступного виразу:

$$G_w(Z_i) = \begin{cases} 1, Z(t_i) = Z(t_{i-1}) \oplus Z(t_{i-1}) = 1; \\ -1, Z(t_i) = Z(t_{i-1}) \oplus Z(t_{i-1}) = 0, \end{cases}$$

де w і z визначаються з двійкових розкладів:

$$w = \sum_{i=0}^{n-1} w_i \cdot 2_{n-1-i}; \qquad Z = \sum_{i=0}^{n-1} Z_i \cdot 2_{n-1-i},$$

де w-індекс, а $\sum_{i=0}^{n-1} w_i$ - вага індексу, або ранг функції Галуа.

Визначимо головні властивості функцій Галуа:

1) повнота і ортогональність

Множина 2^n-1 функцій Галуа утворює повну квазіортогональну систему функцій у просторі дискретно-постійних функцій на інтервалі [0, 2^n-1], причому повна ортогональність існує в просторі (0, +2^n-1), тобто:

$$\phi_1(Z) = \begin{cases} 2^n - 1, при \quad i = j; \\ -1/N, при \quad i \neq j. \end{cases}$$

Дана система функцій Галуа ортогональна у просторі (0, 2^n-1) і ортогональна у просторі (0, -2^n-1), тобто при N=2^n-1→∞, 1/N→0. Такі функції Галуа породжують псевдовипадкові послідовності максимальної довжини (так званні М-сигнали) у просторі (-1, +1) та відомі коди Баркера у просторі (-1, 0, +1).

При цьому ортогональні властивості таких функцій Галуа які характеризуються особливими автокореляційними функціями широко використовуються у системах передавання даних для виявлення, приймання та виправлення помилок у сучасних комп'ютерних мережах та систем.

Наприклад, для 7-ми бітної рекурентної функції Галуа $\phi_1(Z_i) = (-1-1-1+1+1-1+1)$ на інтервалі N=2^3-1 на основі її циклічних зсувів у нормованому (-1,+1) та логічному (0, +1) просторах утворюються відповідні базисні матриці (рис.8.1). Графіки автокореляційних характеристик таких функцій Галуа для розглянутого прикладу розраховані на основі знакового аналітичного виразу $Bxx(j) = \frac{1}{N} \sum_{i=0}^{N-1} signZ_i \otimes signZ_{i+j}$ і приведені на рис.8.2.

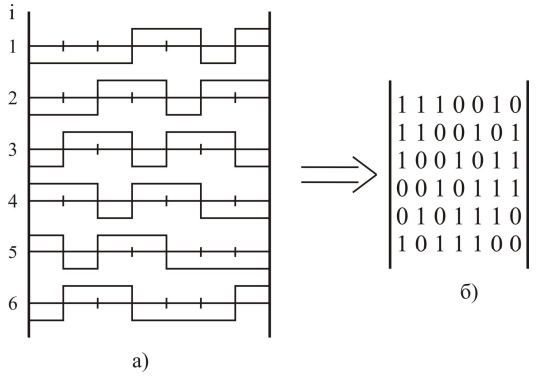


Рис. 8.1. Базисні функції Галуа у нормованому (а) також логічному (б) просторах.

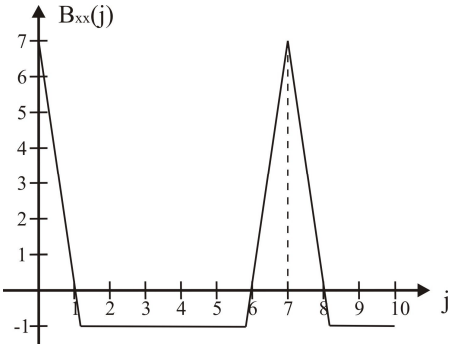


Рис. 8.2. Автокореляційна характеристика базисної функції Галуа на інтервалі 2^n-1 (n=3).

З рис.8.2. видно, що повнота ортогональності таких базисних функцій Галуа однозначно забезпечується у просторі (0, +N), що саме і використовується у телекомунікаційних системах.

Обмеженням таких функцій Галуа є те, що вони породжують тільки унітарні коди Хаара-Галуа згідно кодової матриці (рис. 8.3), де G_1 G_2….G_i….G_{n-1} – біти коду Галуа, G_i є $\overline{0,t}$.

$$\begin{vmatrix} G_1 & 0 & 0 & ..... & 0 & 0 \\ 0 & G_2 & 0 & ..... & 0 & 0 \\ 0 & 0 & G_3 & ..... & 0 & 0 \\ ... & .... & ..... & ..... & .... & .... \\ 0 & 0 & 0 & ..... & 0 & G_{N-1} \end{vmatrix}$$

Рис. 8.3. Кодова матриця базису Хаара-Галуа.

Множина 2^n функцій Галуа утворює повну квазіортогональну систему у просторі нормованих дискретно-постійних функцій на інтервалі $[0, 2^n]$, при чому повна ортогональність існує у просторі $(0,+2^n)$, тобто:

$$\phi_2(Z) = \begin{cases} 2^n, при \quad i = j; \\ -1/N_i, при \quad i \neq j, \end{cases}$$

де значення N_i залежить від інтервалу 2^n та незвідного поліному, на основі якого генерується функція Галуа.

Слід зауважити, що функції $\phi_1(Z)$ та $\phi_2(Z)$ можуть генеруватись одним і тим самим ключем незвідного полінома поля Галуа. При цьому $\phi_2(Z)$ буде відрізнятись від $\phi_1(Z)$ виконанням так званої операції біт-стаффінгу (вставки символу «0») шляхом розширення інтервалу 2^n-1 функції $\phi_1(Z)$ до інтервалу 2^n у діапазоні максимальної кількості її одиничних значень.

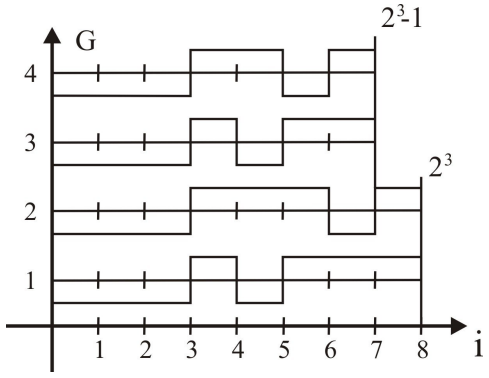


Рис. 8.4. Приклад розширення інтервалу функцій Галуа.

Наприклад, семибітні функції Галуа $\phi_1(Z)$ і $\phi_1(Z)$, які генеруються відповідно різними ключами $G_i+1=G_i \oplus G_i-3$ та $G_i+1=G_i \oplus G_i-2$ на інтервалі 2^3-1 розширюються до однієї функції $\phi_2(Z)$ на інтервалі 2^3 (рис.8.4.)

При цьому породжується система квазіортогональних функцій Галуа у нормованому та логічному просторах, матриці яких показані на рис. 8.5.

Автокореляційна характеристика функцій $\phi_2(Z)$ показана на рис. 8.5, звідки видно, що система таких базисних функцій також ортогональна у просторі $(0, +2^n)$.

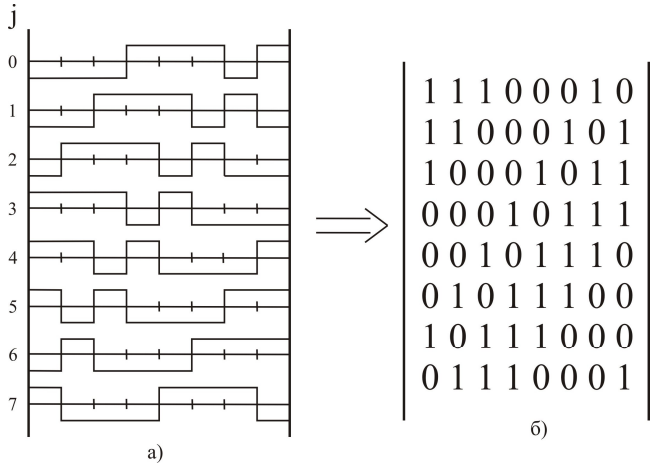


Рис. 8.5. Базисні функції Галуа у нормованому (а) та логічному (б) просторах.

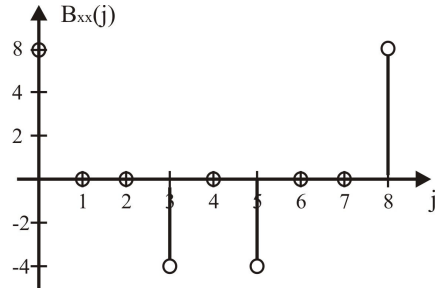


Рис. 8.6. Автокореляційна характеристика базисної функції Галуа на інтервалі 2^n(n=3).

Можна показати, що від'ємні пелюстки автокореляційної характеристики базисної функції Галуа асимптотично зменшуються при розширенні інтервалу 2^n.

Наприклад, код функції Галуа *GF(4/2)* (1111010110010000) має автокореляційну характеристику приведену на рис. 8.7.

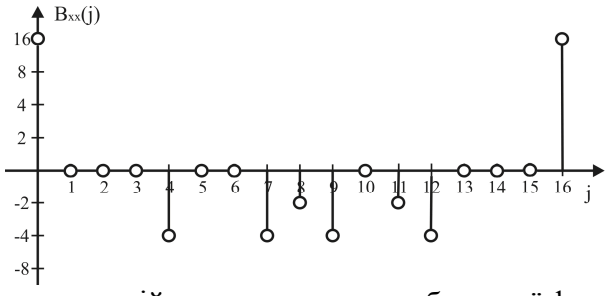


Рис. 8.7. Автокореляційна характеристика базисної функції Галуа на інтервалі 2^3.

У загальному випадку функція ϕ(Z), яка описує початкову систему *P*-ічних функцій алгебри логіки від n-аргументів, є дискретно-постійною на інтервалі [0, p^n-1]. Якщо функція ϕ(Z) така, що:

$$\sum_{Z=0}^{p^n-1}\phi(Z)\cdot G_w(Z)=0;\ \ \omega\epsilon 0,\ \ p^n-1,$$

то ϕ(Z)=0.

Таким чином, функції Галуа є базисом асимптотично ортогональних рядів дискретно-постійних функцій, які описують системи булевих функцій.

Кінцевість ряду, який описує функцію Галуа визначається виходячи з виразів:

$$\phi(Z)=\sum_{\omega=0}^{2^n-1}S(\omega)\cdot G_\omega(Z), \qquad (8.1.)$$

$$S(\omega)=2^{-n}\sum_{Z=0}^{2^n-1}\phi(Z)\cdot G_\omega(Z) \qquad (8.2.)$$

де вираз (8.1.) розклад функцій в ряд Фур'є, а (8.2.) - коефіцієнтами Фур'є.

З (8.1.) слідує, що розклад в ряд по функціях Галуа будь-якої дискретно-постійної функції, що описує деяку систему булевих функцій від n-аргументів, містить не більше 2-х членів, а вага індексів w функцій Галуа у розкладі не перевищує n.

Симетрія індексу і аргументу функції Галуа визначається згідно аксіоми: для будь-яких $\omega,\ Z,\ \epsilon\ \overline{0,2^n}$ справедливо $G_\omega(Z)=G_Z(\omega),$ що слідує з (8.1).

Властивість зсуву аргументу і рекурентність функцій Галуа визначається наступним чином: для будь-яких $\omega,\ Z,\ \tau\ \epsilon\ \overline{0,2^n}$ тоді $G_\omega(Z+\tau)=G_\omega(Z)\cdot G_\omega(\tau)\bmod 2,$ отже, множина функцій Галуа замкнена у полі відносно операції множення.

Властивість рекурентності є особливо важливою у сфері застосування функцій базису Галуа, а також кодів та систем числення в полях Галуа при рішенні широкого класу задач теорії чисел, кодування, перетворення та цифрового опрацювання сигналів, стиснення даних, їх захисту від помилок та несанкціонованого доступу шляхом шифрування.

Використання властивості рекурентності аргументу базису Галуа в наш час склало фундаментальну основу розробки теорії та принципово нових технічних рішень багатьох складних задач і практичних застосувань у галузі цифрового формування, передавання і опрацювання інформаційних потоків, в тому числі:

1) побудову
– кодових систем Галуа;
– генераторів кодів поля Галуа;
– кодових шкал Галуа;
– АЦП Галуа;
– Багатопортової асоціативної пам'яті Галуа;
2) розробки нових методів:
– передавання інформації в умовах інтенсивних завад;
– формування та передавання інформації на основі вертикальної інформаційної технології;
– стиснення даних у базисі Галуа;
– блокової синхронізації потоків даних на основі протоколів Галуа;
– виявлення та виправлення помилок в потоках даних сигнальними коректуючими кодами Галуа;
– рандомізації гармонічних сигналів в кодах Галуа;
– ідентифікації передаварійних та аварійних станів квазістаціонарних об'єктів управління;
– структуризації інформаційних даних в комп'ютеризованих мережах та системах.

8.2. Класи кодів поля Галуа (КПГ).

8.2.1 Одновимірні дворівневі КПГ.

Принципи кодування методом залишків базуються на теоретико-числовому підході і представляють собою клас векторних багаторівневих кодів поля Галуа. Застосування цього класу КПГ дозволяє зменшити надлишковість вимірювальної інформації за рахунок виключення з інформаційного потоку кодів рангів Діофантового рівняння.

$$Y_i = b_i (\mathrm{mod}\, P),$$

яке відповідає лінійному рівнянню з розв'язком у цілих числах

$$Y_i = a_i \cdot P + b_i, \qquad (8.3)$$

де $a_i = \breve{E}\left[\frac{Y_i}{P}\right]$ - ранг цифрового відліку Y_i; $0 \le b_i \le P-1$ - найменший невід'ємний залишок Y_i по модулю P, тобто

$$b_i = rez Y_i (\mathrm{mod}\, P).$$

З виразу (8.3) видно, що величина Y_i може бути однозначно відновлена, якщо крім b_i можна обчислити ранг a_i на основі попередньо відомого Y_{i-1}. Тобто існує рішення функціонеру:

$$Y_i = F(Y_{i-1}, b_i).$$

Покажемо, що достатньо простим при цьому є рекурентний алгоритм обчислення рангу a_i по текучому значенню b_i і відомому рангу a_{i-1} попереднього відліку Y_{i-1}.

Для доведення існування функціонала Y_i рішимо рівняння (8.3) відносно a_i

$$a_i = \frac{Y_i - b_i}{P},$$

Враховуючи цілочисельний характер рангу a_i, введемо праву частину останнього виразу під знак цілочисельної функції

$$a_i = E\left[\frac{Y_i - b_i}{P}\right]$$

Виразимо значення відліку Y_i через результат попереднього вимірювання Y_{i-1}.

$$Y_i = Y_{i-1} + \alpha P \tag{8.5}$$

де α – деяка величина, які визначає відмінність відліків Y_i та Y_{i-1}.

Підставивши (8.5) у a_i і виконавши прості перетворення отримаємо

$$a_i = E\left[\frac{Y_{-1} - b_i}{P} + \alpha\right], \tag{8.6}$$

де α має зміст помилки обчислення рангу відліку Y_i значення Y_{i-1}.

Ранг відліку Y_i з рівняння (8.6) може бути точно визначений виходячи з властивостей цілочисельної функції $E[\cdot]$, якщо величина помилки його обчислення менша одиниці, тобто $0 \le \alpha < 1$.

Доведемо ствердність формули (8.6) при умові, що зміна рангу a_{i-1} та a_{i-} не перевищує одиниці.

$$(a_i - a_{i-1}) \le 1 \tag{8.7}$$

На рис.8.8 показані варіанти зміни цифрових відліків Y_i та Y_{i-1}. при виконанні умови (8.7).

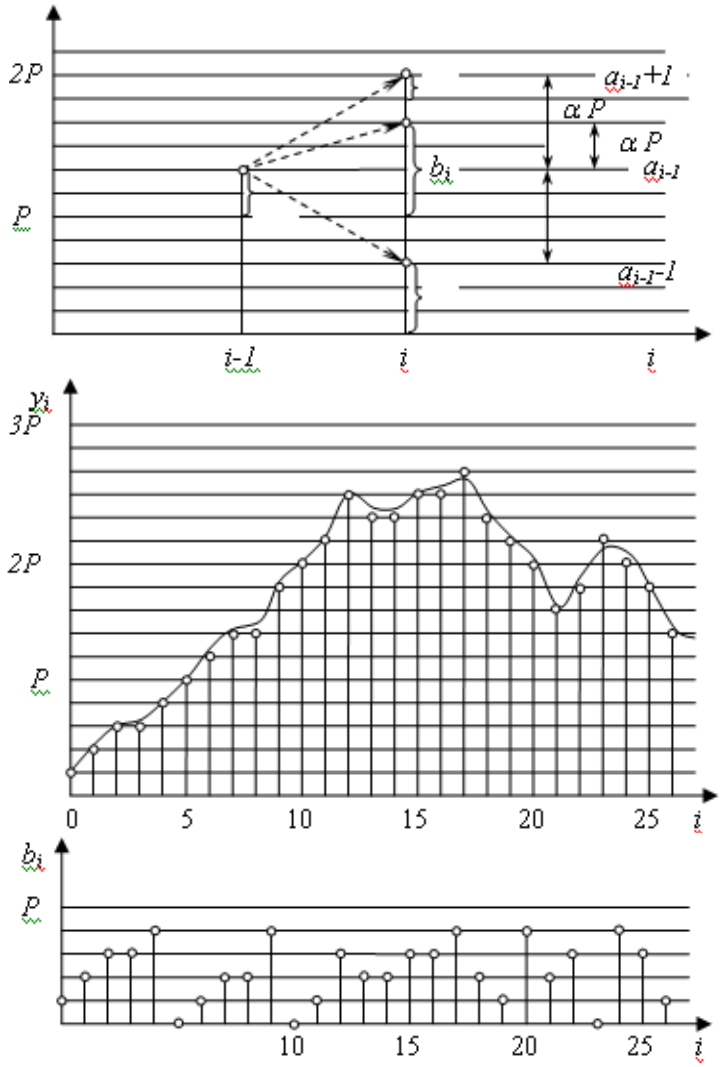


Рис.8.8. Рекурентне кодування інформаційних потоків методом залишків.

Нехай відлік Y_{i-1} має ранг a_{i-1} і залишок b_{i-1}.

Згідно рис.8.8 можливі три випадки отримання відліків, які відрізняються значенням рангів a_i від рангу a_i відліку Y_{i-1}.:

1) Ранг Y_i збільшується на одиницю $(a_i = a_{i-1} + 1)$.

Очевидно, що в цьому випадку приріст відліку Y_i не може бути менший P-b_{i-1}, а також не перевищує величини αP, тобто:

$$P - b_{i-1} \le P + b_{i-1} - b_i \le \alpha P ,$$

або підсилюючи нерівність

$$P - \alpha P \le b_{i-1} - b_i \le b_{i-1}. \qquad (8.8)$$

Підставивши Y_{i-1}.у рівняння (8.6) і потім у (8.4), отримаємо рішення для шуканого Y_i

$$Y_i = \breve{E}\left[a_{i-1} + \frac{b_{i-1} - b_i}{P} + \alpha \right] P + b_i. \qquad (8.9)$$

З співвідношень (8.8) та (8.9) витікає оцінка для текучого відліку у вигляді нерівності

$$E\left[\alpha_{i-1} + \frac{P - \alpha P}{P} + \alpha \right] P + b_i \le Y_i \le E\left[a_{i-1} + \frac{b_{i+1}}{P} + \alpha \right] P + b_i,$$

звідси враховуючи, що $b_{i-1} < P$, отримаємо

$$E[a_{i-1} + 1]P + b_i \le Y_i \le E[a_{i-1} + 1 + \alpha]P + b_i \qquad (8.10)$$

Верхня границя у (8.10) співпадає з нижньою, оскільки $\alpha < 1$ *і* $E[a_{i-1} + \alpha + 1] = [a_{i-1} + 1]$.

Тому

$$Y_i = E[a_{i-1} + 1]P + b_i$$

і справедливість формули (8.6) доведена для випадку 1).

2) ранг по відношенню до Y_{i-1}. зберігається $(a_i = a_{i-1})$.

Використовуючи рівняння (8.9) та нерівність

$$-\alpha P \le b_{i-1} - b_i \le \alpha P,$$

які слідують з рис.8.8, оцінимо, як і у випадку п.1) значення Y_i

$$E\left[a_{i-1} - \frac{\alpha P}{P} + \alpha \right] P + b_i \le Y_i \le E\left[a_{i-1} - \frac{\alpha P}{P} + \alpha \right] P + b_i,$$

звідки

$$E[a_{i-1}]P + b_i \le Y_i \le E[a_{i-1} + 2\alpha]P + b_i. \qquad (8.11)$$

Границі в (8.11) співпадають, якщо

$$2\alpha = \frac{2(b_{i-1} - b_i)}{P} < 1 \quad або \quad \frac{b_{i-1} - b_i}{P} < 0.5 \qquad (8.12)$$

і формула (8.6) доведена для випадку 2).

3) Ранг Y_i зменшується на одиницю $(a_i = a_{i-1} - 1)$.

Приріст відліку Y_i згідно рис.8.8 у цьому випадку не менше залишку b_{i-1} і не перевищує αP.

Оцінюючи, як і у попередніх випадках значення Y_i

$$E\left[a_{i-1} - \frac{b_{i-1} - P}{P} + \alpha \right] P + b_i \le Y_i \le E\left[a_{i-1} - \frac{2P - P}{P} + \alpha \right]$$

і підсилюючи оцінку нерівності знизу $b_{i-1} \ge -\alpha P$, отримаємо

$$E[a_{i-1}-1]P+b_i \le Y_i \le E[a_{i-1}-1+2]P+b_i. \qquad (8.13)$$

Підставивши помилку обчислення рангу α в умову (8.12), добиваємося співпадання границь у (8.13). Тим самим доведення формули (8.6) завершено.

З викладеного доведення, видно, що умова існування алгоритму обчислення рангу відліку Y_i на основі Y_{i-1} та b_i по формулі (8.6) виконується, коли для двох послідовних відліків справедлива нерівність

$$-0.5 < \frac{b_{i-1}-b_i}{P} < 0.5 \qquad (8.14)$$

Остання умова також обмежує допустиму величину першої різниці між відліками Y_{i-1}. та Y_i

$$|Y_i - Y_{i-1}| < 0.5P, \qquad (8.15)$$

що добре узгоджується другим випадком, коли залишки b_i та b_{i-1} належать одному рангу $a_i = a_{i-1}$.

Таким чином, рівняння (8.9) задовольняє вимоги однозначного декодування для всіх розглянутих випадків, якщо $\alpha = 0.5$. При цьому у відповідності з (8.6) помилка обчислення рангу a_i завжди буде достатньою.

Враховуючи встановлений знак помилки α у виразах (8.6) та (8.9), отримаємо шукану рекурентну формулу для відновлення потоку вимірювальних даних, які кодуються методом залишків багаторівневим векторним кодом поля Галуа

$$Y_i = \hat{E}\left[\frac{Y_{i-1}+b_i}{P}+0.5\right]P+b_i \qquad (8.16)$$

де $\hat{E}[\cdot]$ - цілочисельна функція з округленням до більшого цілого.

Розглянемо метод кодування даних методом залишків на прикладі рис.8.8.

Кодування ведеться по модулю $P=5$, тому найбільший перший приріст відліків $|Y_i - Y_{i-1}| \le 2$ згідно умови (8.15).

Нехай i=8, тоді: Y_{i-1}=7; b_i=2; b_{i+1}=4; b_{i+2}=0; b_{i+3}=1. Згідно (8.16) маємо:

$$Y_i = \hat{E}\left[\frac{7-2}{5}+0.5\right]5+2 = \hat{E}[1.5]5+2 = 7;$$

$$Y_{i+1} = \hat{E}\left[\frac{7-4}{5}+0.5\right]5+4 = \hat{E}[1.1]5+4 = 9;$$

$$Y_{i+1} = \hat{E}\left[\frac{9-0}{5}+0.5\right]5+0 = \hat{E}[2.3]5+0 = 10;$$

$$Y_{i+1} = \hat{E}\left[\frac{10-1}{5} + 0.5\right]5 + 1 = \hat{E}[2.3]5 + 1 = 11.$$

Таким чином, доведені умови та продемонстрована однозначність кодування та декодування інформаційних потоків багаторівневими векторними кодами поля Галуа.

Найширше застосування такі коди отримали у алгоритмах стиснення інформації та спектрального аналізу сигналів на основі нелінійних цифрових фільтрів.

8.2.2. Двовимірні дворівневі КПГ.

Двовимірні дворівневі КГП можуть бути задані у вигляді матриць з числом елементів $V = 2^k \cdot 2^m$, які приймають пари бінарних значень $b_{ij} \in$ 00, 01, 10, 11; $i \in \overline{1,2^k}$; $j \in \overline{1,2^m}$

$$b_{ij} = \begin{vmatrix} b_{k1} & b_{k2} & \dots & b_{kj} & \dots & b_{km} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{ij} & \dots & b_{im} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2m} \\ b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1m} \end{vmatrix} \qquad (8.17);$$

Приклад двохмірного коду Галуа GG_2^3 ілюструється бульовою матрицею

$$GG\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{vmatrix} 01 & 01 & 01 & 00 & 01 & 00 & 00 & 00 \\ 01 & 01 & 01 & 00 & 01 & 00 & 00 & 00 \\ 01 & 01 & 01 & 00 & 01 & 00 & 00 & 00 \\ 11 & 11 & 11 & 10 & 11 & 10 & 10 & 10 \\ 01 & 01 & 01 & 00 & 01 & 00 & 00 & 00 \\ 11 & 11 & 11 & 10 & 11 & 10 & 10 & 10 \\ 11 & 11 & 11 & 10 & 11 & 10 & 10 & 10 \\ 11 & 11 & 11 & 10 & 11 & 10 & 10 & 10 \end{vmatrix}$$

Таке подання двомірного дворівневого КПГ (2D КПГ) визначає теорію і методологію кодування інформації у двомірному дискретному просторі. При цьому нове рішення отримують у кратному випадку дві задачі:

1) прив’язка координат і позиціювання об’єкта в мережі двомірного простору з двох бітовими мітками Галуа. При цьому у

залежності від геометрії площі позиціювання по вертикалі і горизонталі можуть бути вибрані різні 1D КПГ (рис.8.9).

2)

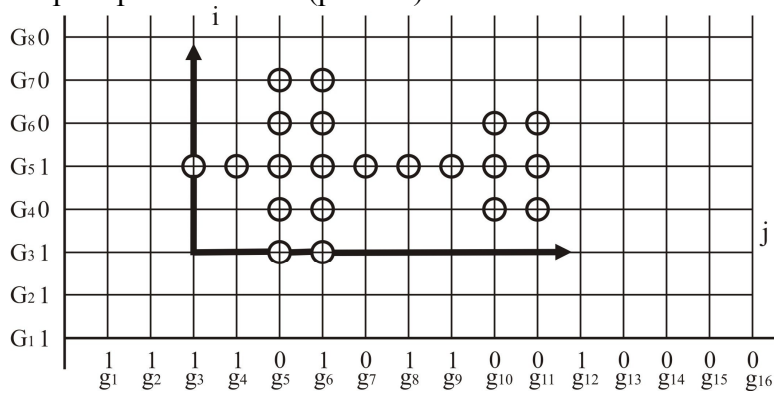


Рис.8.9. Позиціювання об'єкта в координатах 2D коду поля Галуа.

Згідно рис.8.9 однозначне позиціювання об'єкта в координатах 2D КПГ описується наступним кодом базисних функцій нульового порядку

$$F(ij)=\begin{cases} G_7 : g_4\, g_5\, \overline{g}_6\, \overline{g}_7\, g_8 g_9 g_{10} g_{11} g_{12} \\ G_6 : g_4\, g_5\, \overline{g}_6\, \overline{g}_7\, g_8 g_9 g_{10}\, \overline{g}_{11}\, \overline{g}_{12} \\ G_5 : \overline{g}_4\, \overline{g}_5\, \overline{g}_6\, \overline{g}_7\, \overline{g}_8\, \overline{g}_9\, \overline{g}_{10}\, \overline{g}_{11} g_{12} \\ G_4 : g_4\, g_5\, \overline{g}_6\, \overline{g}_7\, g_8 g_9 g_{10}\, \overline{g}_{11}\, \overline{g}_{12} \\ G_3 : g_4\, g_5\, \overline{\overline{g}_6\, \overline{g}_7}\, g_8 g_9 g_{10} g_{11} g_{12} \end{cases} \qquad (8.18)$$

Вираз (8.18) подамо у вигляді мулевих значень $G_i g_j$:

$$F(ij)=\begin{cases} 0 : 1\,0\,0\,1\,1\,1\,0\,0\,1 \\ 0 : 1\,0\,0\,1\,1\,1\,0\,1\,0 \\ 1 : 0\,1\,1\,0\,0\,0\,1\,1\,0 \\ 0 : 1\,0\,0\,1\,1\,1\,0\,1\,0 \\ 1 : 1\,0\,0\,1\,1\,1\,0\,0\,1 \end{cases}$$

Аналіз такого способу застосування 2D КПГ показує, що при стандартному кодуванні об'єкта у базисі Радемахера потрібно код з ентропією

$$I_x = \left(\log_2 k + \log_2 m\right)\cdot V\,,$$

де V-число точок ідентифікації координат об'єкта .

Для розглянутого прикладу:

$$I_x(R) = \left(3+4\right)\cdot 21 = 147 біт\,;$$

$$I_x(G) = 5\cdot 10 = 50 біт\,,$$

Що практично у три рази менше у базисі Галуа по відношенню до базису Радемахера.

Оцінка зменшення об'єму інформації при кодуванні зображень у базисі Галуа аналітично описується наступним виразом:

$$k_{еф} = \frac{V(\log_2 n + \log_2 m)\cdot \log_2 a}{n_j \log_2 a \cdot m_j} \qquad (8.19)$$

де n_j, m_j – відповідно число інформаційних елементів 2D КПГ, в паралелепіпеді яких вміщений кодований об'єкт; α – число рівнів квантування елементів півтонового зображення. З виразу (8.19) очевидно, що ефективність кодування зображень у базисі Галуа не залежить від значення α.

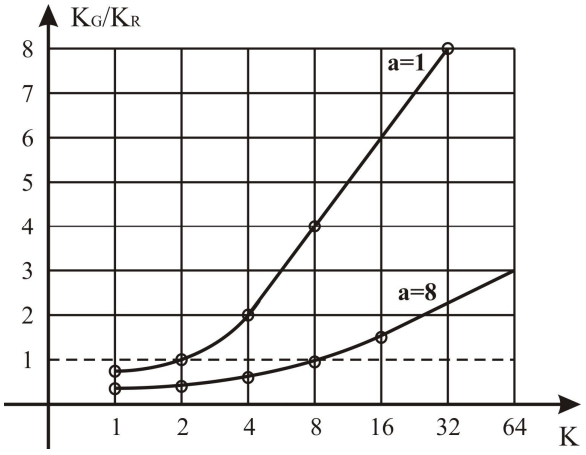


Рис.8.10. Відносна ефективність кодування зображень на основі 2D КПГ.

На рис.8.10 показані графіки відносної ефективності кодування дворівневих зображень.

8.2.3. Просторова форма двовимірних КПГ.

Просторова форма КПГ на основі розгалуженого кодового графа.

Така форма КПГ описується математичною моделлю:

$$N_j = (b_0\ b_1\ ...b_j);$$

$$b_j = res(b_{j-1} + b_{j+n}) \bmod P. \qquad (8.20)$$

Це подання КПГ визначає методологію побудови транспортних систем і мережу маршрутизації мобільних об'єктів на площині. При цьому реалізація «м'якого» позиціювання зводиться до розміщення вздовж розгалужених транспортних шляхів сигнальних міток КПГ, маніпульованих

згідно деяких двох ознаках: символах, кольору, геометричному розмірі, електричному, оптичному, магнітному та інш. параметру чи ефекту.

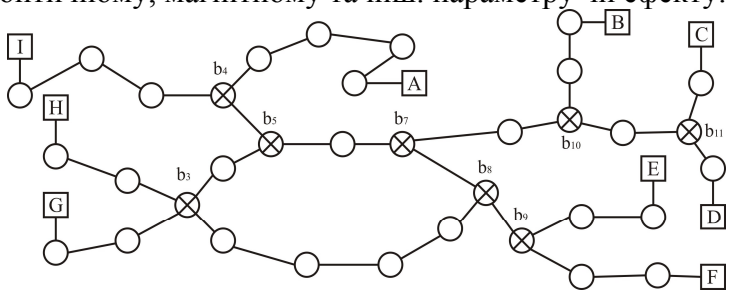


Рис.8.11. Структура транспортної мережі руху мобільного об’єкта в двох бітовому КПГ.

На рис.8.11 показаний приклад розгалуженої структури транспортної мережі, в якій елементи позиціювання утворює КПГ кільця $GF\begin{pmatrix}4\\2\end{pmatrix} = (1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0)$.

У процесі переміщення об’єкта послідовно зчитується сигнальні мітки двох бітового КПГ. Причому *n* – розрядний кодон визначає місцезнаходження об’єкта на траєкторії руху. Дії об’єкта руху, повертання вправо або вліво визначається виходячи з текучого кодона КПГ і програми руху, яку об’єкт отримує на початковому або кінцевому пункті.

Для однозначного позиціювання місцезнаходження і програмного переміщення рухомого об’єкту між різними пунктами A, B, C, D, E, F, G, H, L сила дається програма, яка в операторній формі має вигляд:

$$S_1 : (b,...b_i) \equiv V_{1j}(b_i,...,b_j) \equiv V_{ij}...(b_m,...,b_k) \equiv S_2,$$

де: S_1, S_2 – відповідні символи (коди) початкового та кінцевого пунктів переміщень, V_j – ознака повороту:

$$V_j = \begin{cases} 0, \uparrow (вліво) \\ 1, \downarrow (направо) \end{cases},$$

Наприклад програми можливих шляхів переміщень рухомого об’єкта (РО) з пункту Н в пункт Е має наступний вид:

$$1)H : b_1\ b_2\ b_3 \equiv 0;\quad b_3\ b_4\ b_5 \equiv 1;\quad b_5\ b_6\ b_7 \equiv 1;\quad b_7\ b_8 \equiv 0;\quad b_8\ b_9 \equiv 0;\quad b_9\ b_{10}\ b_4 \equiv 0$$

$$2)H : b_1\ b_2\ b_3 \equiv 1;\quad b_3\ b_4\ b_5\ b_6\ b_7\ b_8 \equiv 1;\quad b_6\ b_9 \equiv 1;\quad b_9\ b_{10}\ b_{11} \equiv 0;$$

Описана форма двох бітового КПГ не є реверсивною, тому транспортна мережа на її основі характеризується одно направленістю, тобто переміщення РО можливі тільки тільки з пунктів A, G, H, L в один з інших B, C, D, E, F.

Для опису програмного руху РО у зворотньому напрямі необхідно подати модель(М) в інвертованому виді:

$$N_j = (b_j, b_{j-1} \ldots b_0)$$

$$b_{j-1} = res(b_j + b_{j+n}) \bmod P,$$

А у програму руху ввести ознаку зворотнього руху. Тоді:

$$V_j = \begin{cases} S_0, \rightarrow; \\ 0, \uparrow; \\ S_1, \leftarrow; \\ 1, \downarrow. \end{cases}$$

При цьому для правильного зчитування КПГ у зворотньому напрямі РО повинен переміщуватися на *n-1* позицію дальше мітки початку зворотнього руху або завантажитись зворотнім ключем коду Галуа.

Приклад програми руху РО з пункту А в пункт Н має вигляд:

$$\begin{aligned} A: b_m b_1\ b_2\ b_3\ b_4 &\equiv 0; \\ b_4\ b_5 &\equiv 1; \\ b_5\ b_6\ b_7\ b_8 &\equiv S; \\ b_8\ b_7 &\equiv 1; \\ b_7\ b_6\ b_5 &\equiv 0; \\ b_5\ b_4\ b_3 &\equiv 1; \\ b_3\ b_2\ b_1 &\equiv: H; \end{aligned}$$

Очевидно, що розширивши число ознак V_i наприклад:

$$V_i = \begin{cases} S_0, \rightarrow; \\ 0, \uparrow; \\ 1, \downarrow \\ S_1, \lrcorner \\ S_2, \urcorner \\ S_3, \leftarrow \end{cases}$$

Де символи $\uparrow, \downarrow$ - відповідно визначають рух вгору або вниз, можна побудувати багаторівневу 3D мережу позиціювання переміщень РО в трьохмірному просторі.

8.2.4. Полярно-спіральна форма КПГ.

Полярно-спіральна форма КПГ у вигляді двомірної розгортки описується матрицею:

$$N_{ij} = \begin{vmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{ij} & \dots & b_{im} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nj} & \dots & b_{nm} \end{vmatrix}, \qquad (8.21)$$

де:

$$b_{ij} = res(b_{ij} + b_{i,j+m}) \bmod \rho$$
$$b_{ij} = res(b_{i-1,j} + b_{i+n,j}) \bmod P$$

Для КПГ $GF(2^4)$ матриця (8.21) має вигляд:

$$N_{ij} = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & b_{ij} & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

Особливістю такого подання КПГ є одночасне існування рекурентності по спіралі та твірній циліндра. Дана властивість спіральних форм КПГ ефективно використана в сигнальних коректуючи кодах Галуа для виявлення та виправлення пачок помилок в каналах зв'язку сучасних безпровідних сенсорних та комп'ютерних мережах.

Нехай маємо код Галуа G_2^4 який формується рекурентним кодом

$$G_{i+1} = G_i \oplus G_{i-4}, \qquad (8.22)$$

тобто

$$G_2^4 = 1111\ 01011\ 0010\ 0011\ 1101\ 0110$$
$$0100\ 0111\ 1010\ 1100\ 1000\ 1111\ 0101\ \dots \qquad (8.23)$$

Можна показати, що код генерований на основі виразу (8.23) можна упакувати в спіраль (рис.8.12), причому по кожній з чотирьох твірних

формується рекурентна послідовність, яка має відповідні рекурентні властивості коду в базисі Галуа $G_{i+1} = G_i \oplus G_{i-4}$.

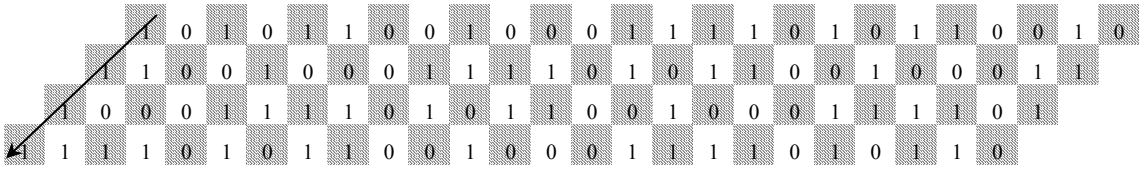


Рис. 8.12. Сигнальний коректуючий код упакований у вигляді спіралі.

Але коли дану спіраль закодовану рекурентним кодом розкрутити, то рекурентність збережеться через 12 символів і можливо буде виправляти помилки по твірних спіралі згідно виразів:

$$G_{i+1} = G_i \oplus G_{i-13}. \qquad (8.24)$$

Виправлення помилок в сигнальних кодах з врахуванням спіральних властивостей відбувається наступним чином: з врахуванням спіральних властивостей сигнальний рекурентний код можна використати при виявлені пакетів помилок. Оскільки спочатку виконується перевірка і виправлення помилок згідно виразу (8.22), а потім, згідно виразу (8.24) - по твірних спіралі.

На рис.8.13 показано виявлення помилок при використанні спіральних властивостей сигнальних коректуючих кодів поля Галуа. Як видно з рис. 8.13, при виникненні помилок в п'яти підряд позиціях, виявлення помилок завдяки рекурсивному виразу (8.23) стає неможливим, оскільки можлива неправильна корекція. Тому потрібно виявляти дані помилки завдяки спіральним властивостям коду Галуа, тобто виразу (8.24).

Номер позиції бітів	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	…
G_4^2	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	…
G_{i-4}										*	*	*	*	*									
G_{i-13}													*										

Рис. 8.13. Виявлення помилок при використанні спіральних властивостей сигнальних коректуючих кодів поля Галуа.

8.2.5. Двомірний однобітовий КПГ.

Даний клас кодів Галуа формується на основі квадратної матриці з першою стороною і першим стовпцем у вигляді лінійних рекурентних послідовностей з ключем

$$G_{i+1} = a_i G_i \oplus a_{i-1} G_{i-1} \oplus \cdots \oplus a_{i-n} G_{i-n},$$

де $a_i \in \overline{0,1}$ для всіх бітів Галуа стартового коду $(G_i, G_{i-1}, ..., G_{i-n})$.

У загальному випадку матриця коду Галуа даного класу має вигляд

$$\begin{vmatrix} G_i & G_{i-1} & \dots & G_j & \dots & G_{i-n} \\ G_{i-1} & G_{i-2} & \dots & G_{j-1} & \dots & G_i \\ G_{i-2} & G_{i-3} & \dots & G_{j-2} & \dots & G_{i-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ G_{i-n} & G_i & \dots & G_{j+1} & \dots & G_{i-n+1} \end{vmatrix}$$

Наприклад: для поля Галуа $GF(\binom{3}{2})$ з ключами

$G_{i+1} = G_i \oplus G_{i-3}$ та $G_{i+1} = G_i \oplus G_{i-2}$ - відповідно отримуємо матриці двомірних кодів Галуа даного класу

$$\begin{vmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}; \qquad \begin{vmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{vmatrix}$$

Аналіз приведених двомірних кодів Галуа показує, що вони характеризуються симетрією, як видно з рис 8.14 і потужними можливостями виявлення та виправлення помилок.

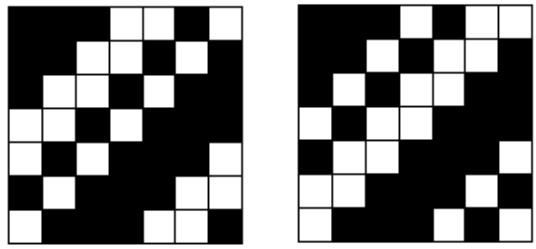


Рис 8.14. Симетрія двомірних кодів поля Галуа.

Їх застосування у системах передавання даних в умовах впливу інтенсивних завад або значному затуханні сигналів у каналах зв'язку.

РОЗДІЛ 9

ТЕОРЕТИКО-ЧИСЛОВІ БАЗИСИ ТА СИСТЕМИ ЧИСЛЕННЯ НА ОСНОВІ СУМ ПОЛІВ ГАЛУА

9.1. Теоретичні основи цілочисельної СЗК базису Крестенсона.

Найбільш загальною математичною основою ТЧБ є базис прямих ортогональних сум полів Галуа. Даний базис визначається властивостями Китайської теореми про залишки і описується системою дискретних ортогональних функцій Віленкіна-Крестенсона у системі взаємно простих модулів $P_1, P_2 \ldots P_k$. Цей базис породжує систему числення залишкових класів (СЗК), яка представляє коди поля Галуа у k-мірному просторі.

Теоретично основа СЗК глибоко досліджується І.Я. Акушським, В. М. Амербаєвим, Г.Є. Пуховим, В.М. Синьковим, В.А. Торгашевим, Є.І. Брюховичем, М.І. Черв'яковим, А.А. Колядою, а також автором.

Теоретичною базою цілочисельного перетворення СЗК є доведення існування в кільці набору сукупності елементів $y_1, y_2, \ldots y_j, \ldots, y_k$, які задовільняють систему Діафантових рівнянь (порівнянь)

$$\left.\begin{array}{c} y_j \equiv 1(\mathrm{mod} A_j) \\ y_j \equiv 0(\mathrm{mod}\prod^{k} A_j), \forall_j = \overline{1,k} \end{array}\right\},$$

де A_j – символ ідеалу причому

$$A_j + \prod_{i \neq j}^{k} A_i = A; \forall_j = \overline{1,k}.$$

В СЗК елементи Y_j позначається через B_j і називаються ортогональними базисами. При цьому $\{B_j\}$ повинні зодовільняти систему порівнянь

$$\left.\begin{array}{c} \sum_{j=1}^{k} B_j \equiv 1(\mathrm{mod}\prod_{j=1}^{k} P_j); \\ B_j \equiv 1(\mathrm{mod} P_j) \end{array}\right\}, \tag{9.1}$$

де P_j – попарно взаємнопрості модулі, звідки слудує, що:

$$\left.\begin{array}{c} B_j \equiv 1(\mathrm{mod}\prod_{j=1}^{k} P_j; \\ B_j \equiv A(\mathrm{mod} P_j), i \neq j, \forall = \overline{1,k} \end{array}\right\}. \tag{9.2}$$

Рішення (9.2) означає, що кожне B_j ділиться без остачі на всі P_i, якщо $i \equiv j$ і кожне B_j ділиться без остачі на добуток P_j, тобто

$$B_j = m_j \prod_{i \neq j}^{k} P_i,$$

де m_j- деяке ціле число у діапазоні $1 \le m_j \le P_j - 1$.

При цьому діапазон однозначного представлення чисел N_j у цілочисельних СЗК знаходиться у межах

$$0 \le N_j \le \prod_{j=1}^{k} P_j - 1.$$

Якщо позначити

$$P = \prod_{j=1}^{k} P_j \text{ , то } B_j = m_j \frac{P}{p_j} \tag{9.3}$$

А m_j може бути знайдено з рішення Діафантового рівняння

$$m_j \frac{P}{p_j} \equiv 1 (\text{mod } P_j). \tag{9.4}$$

Викладенні теоретичні положення визначають пару перетворень цілочисельної СЗК у вигляді зворотнього

$$N_j = res \sum_{j=1}^{k} b_j B_j (\text{mod } P) \tag{9.5}$$

та прямого перетворення

$$b_j = res N_k (\text{mod} P_j); \forall_j = \overline{1,k}, \tag{9.6}$$

де b_j – найменший невідємний залишок числа N_j по модулю P_j, тобто $0 \le b_j \le P_j - 1$.

Розглянемо приклад:

Нехай задана система взаємно простих модулів: P_1=3, P_2=5, P_3=7.

Визначимо діапазон однозначного кодування чисел P згідно виразу (9.3)

$$P = 3 \cdot 5 \cdot 7 = 105,$$

та ортогональні базиси B_j

$$B_1 = \frac{105}{3} m_1 \equiv 1(\text{mod}3);\ m_1 = 2; B_1 = 70;$$

$$B_2 = \frac{105}{5} m_2 \equiv 1(\text{mod}5);\ m_2 = 1; B_2 = 21;$$

$$B_3 = \frac{105}{7} m_2 \equiv 1(\text{mod}7); \quad m_3 = 1; B_3 = 15;$$

Для перевірки правильності розрахунків задамо: N_j=1 тоді b_1=b_2=b_3=1 згідно виразу (9.5) отримаємо:

$$N_j = res(1 \cdot 70 + 1 \cdot 21 + 1 \cdot 15)(\text{mod } 205) = 1.$$

Представимо розглянуту СЗК у вигляді системи ортогональних функцій Віленкіна-Крестенсона (рис. 9.1).

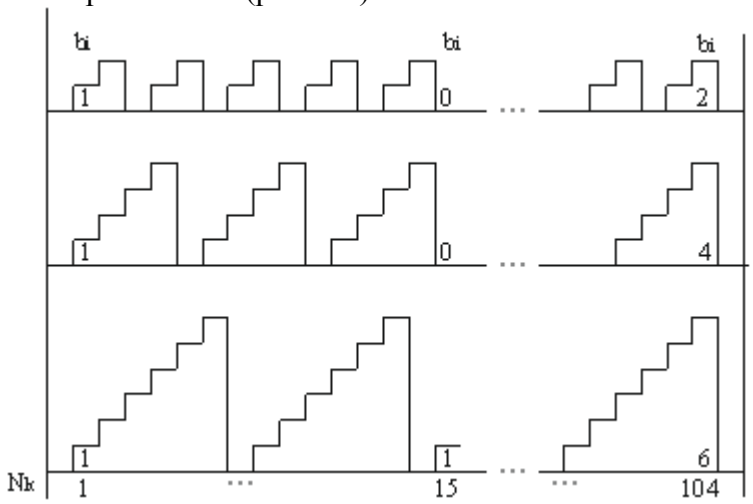


Рис.9.1. Сукупність ортогональних функцій Віленкіна-Крестенсона у системі модулів СЗК: P_1=3; P_2=5; P_3=7.

З рис.9.1 видно розраховане число N_1=1 подано кодом Галуа в СЗК (1, 1, 1) ортогональне базисне число: $B_3 = 15 = (0,0,1)$, а $N_k = P - 1 = 104 = (P_1 - 1, P_2 - 1, P_3 - 1) = (2,4,6).$

Традиційною галуззю застосування такого класу кодів Галуа та цілочисельної форми СЗК є побудова високопродуктивних процесорів, які працюють у модульній арифметиці СЗК.

Особливістю такої арифметики залишкових класів є відсутність наскрізних переносів, які існують у відомій 2-й системі числення базису Радемахера. Тобто: додавання виконується згідно наступного виразу:

$$\begin{array}{l} x = (a_1, a_2, \ldots, a_j, \ldots, a_k) \\ \underline{y = (b_1, b_2, \ldots, b_j, \ldots, b_k)} \\ z = (c_1, c_2, \ldots, c_j, \ldots, c_k) \end{array}, \qquad (9.7)$$

де $c_j = res(a_j + b_j) \bmod P_j,\ j \in \overline{1,k}.$

Подібно без наскрізних переносів виконується операція множення цілих чисел

$$\begin{array}{l} x = (a_1, a_2, \ldots, a_j, \ldots, a_k) \\ \underline{{}^{*}y = (b_1, b_2, \ldots, b_j, \ldots, b_k)} \\ z = (d_1, d_2, \ldots, d_j, \ldots, d_k) \end{array}, \qquad (9.8)$$

де $d_j = res(a_j \cdot b_j) \bmod P_j,\ j \in \overline{1,k}.$

Наприклад: x=13; y=7; (P_1, P_2, P_3)=(3, 5, 7).

Визначимо коди СЗК чисел x та y згідно (9.6):

$$x \Rightarrow \begin{cases} res13(mod3)=1;\ a_1=1; \\ res13(mod5)=3;\ a_2=3; \\ res13(mod7)=6;\ a_3=6; \end{cases} \quad x=(1,3,6).$$

$$y \Rightarrow \begin{cases} res7(mod3)=1;\ b_1=1; \\ res7(mod5)=2;\ b_2=2; \\ res7(mod7)=0;\ b_3=0; \end{cases} \quad y=(1,2,0).$$

Виконаємо операції додавання та множення в кодах СЗК згідно (9.7) та (9.8):

$$\begin{array}{c} P_1\ P_2 P_3 \\ x=(1,\ 3,\ 6) \\ +y=(1,\ 2,\ 0) \\ \hline z=(2,\ 0,\ 6) \end{array}; \quad \begin{array}{c} P_1\ P_2 P_3 \\ x=(1,\ 3,\ 6) \\ \times y=(1,\ 2,\ 0) \\ \hline z=(1,\ 1,\ 0) \end{array}.$$

Перевіряємо результати згідно виразу (9.5):

$$z=(2\cdot 70+0\cdot 21+6\cdot 15)\bmod 105=20;$$

$$D=(1\cdot 70+1\cdot 21+0\cdot 15)\bmod 105=91.$$

В окремих випадках, коли з якихось умов відомо, що $x \geq y$ в СЗК однозначно виконується операція віднімання. Тобто:

$$\begin{array}{c} P_1 P_2 P_3 \\ x=(1,\ 3,\ 6) \\ -y=(1,\ 2,\ 0) \\ \hline V=(0,\ 1,\ 6) \end{array}$$

і $x-y=(0\cdot 70+1\cdot 21+6\cdot 15)\bmod 105=6$

Подібним чином, якщо відомо, що D цілочисельно ділиться на y, то у СЗК можливо виконати операцію ділення шляхом еквівалентного множення згідно виразу:

$$\begin{array}{c} D=(d_1,\ d_2,\ d_3) \\ \div\ Y=(y_1,\ y_2,\ y_3) \\ \hline x=(a_1,\ a_2,\ a_3) \end{array} \qquad (9.9)$$

Наприклад $D=4\cdot 11=44$;

$D=44=(2,4,2)$ тоді y=11=(2,1,4)

$$\begin{array}{l} D=(2,4,2) \\ Y=(2,1,4) \\ \hline x=(1,4,4) \end{array} \qquad \text{тобто:} \qquad \begin{array}{l} P_1(1\cdot 2)\bmod 3=2; \\ P_2(4\cdot 1)\bmod 5=4; \\ P_3(4\cdot 4)\bmod 7=4. \end{array}$$

Використання особливих наборів модулів цілочисельного СЗК у діапазоні кодування чисел Ферма та Мерсена забезпечили ефективну реалізацію швидкого перетворення Фур'є-Галуа і відповідне ефективне застосування ТЧБ в галузі цифрової фільтрації сигналів.

Автором розширена сфера застосування СЗК для формування зменшення надлишковості інформації, підвищення завадозахищеності

передавання даних та побудови високопродуктивних процесорів стосовно задач низових комп'ютерних систем (НКС). Результатом теоретичних досліджень у цій галузі є розробка сукупності нових методів кодування повідомлень, які передаються по каналах зв'язку та вводяться в комп'ютери.

9.2. Кодування інформаційних потоків в СЗК з довільним порядком реєстрації даних.

Суть даного метода полягає у тому, що кожен цифровий відлік Y_{ij}, *i*-го виміру *j*-го каналу множать на базисне число B_j і створюють нове значення $a_{ij} = Y_{ij} \cdot B_j$. Числа a_{ij} сумують по модулю *P* і фіксують у вигляді числа

$$N_{ik} = res \sum_{j=1}^{k} a_{ij} \,(\text{mod } P),$$

де *k*- число каналів.

Таким чином N_k однозначно виражає набір цифових відліків Y_{i1}, $Y_{i2}, \ldots, Y_{ij}, \ldots . Y_{ik}$. При цьому однозначність забезпечується коли $0 \le Y_{ij} = A_j \le P_j - 1$, де: A_j- діапазон квантування цифрових відліків у *j*-му каналі, P_j- модулі перетворення СЗК.

Однозначне відновлення відліків Y_{ij} після передавання або збереження коду N_{ik} виконується на основі прямого перетворення в СЗК.

$$Y_{ij} = res N_{ik} \,(\text{mod } P_j).$$

Коєфіцієнт стиснення інформації згідно оцінки міри Хартлі визначається згідно відношення ентропії вхідних даних, де код кожного цифрового відліку з довільним порядком реєстрації супроводжується іденттифікаційним кодом номера каналу $k_c = \dfrac{I_{âõ}}{I_{ÑÇÊ}}$.

Тобто:

$$I_{âõ} = \sum_{j=1}^{k} \hat{E}\left[\log_2 A_j\right] + k\hat{E}\left[\log_2 k\right];$$

$$I_{âèõ} = \hat{A}\left[\log_2 (P-1)\right].$$

Чисельне моделювання описаного методу пакування даних на основі кодів Галуа цілочисельної СЗК показало, що значення.

$$\sum_{j=1}^{k} \hat{E}\left[\log_2 A_j\right] \approx \hat{E}\left[\log_2 (P-1)\right] \pm 1.$$

Таким чином ефект стиснення даних визначається згідно виразу:

$$k_c = 1 + k\hat{E}\left[\log_2 k\right] / \hat{E}\left[\log_2 (P-1)\right].$$

При зростанні числа каналів оцінка коефіцієнта стиснення інформації асимптотично наближається до величини 1,8.

Наприклад:

$0 \le Y_{ij} \le 10;\ j \in 4,$ тоді $P_j = (10,11,12,13)$ і $P-1 = 17159$

$$k_c = 1 + 4 \cdot \frac{2}{15} = 1 + 0{,}53 \approx 1.5.$$

На основі аналізу характеристик та функціональних можливостей використання ТЧБ для формування структуризованих даних (СД) на низових рівнях РКС розроблений метод компактного завадозахищеного формування СД на основі ТЧБ Крестенсона, який описується наступним алгоритмом:

$$\left.\begin{array}{l} x_1(t) \to x_{i1} \to p_1 \to b_1 \\ x_2(t) \to x_{i2} \to p_2 \to b_2 \\ \ldots \\ x_j(t) \to x_{ij} \to p_j \to b_j \\ \ldots \\ x_m(t) \to x_{im} \to p_{k-1} \to b_{k-1} \\ D_i \to p_k \to b_k \end{array}\right\} N_k = res \sum_{j=}^{k} b_j B_j (\mathrm{mod} P) \qquad (9.10)$$

$$B_j = \frac{P}{p_j} m_j = 1 (\mathrm{mod}\ P_j),$$

де $x_j(t)$ - аналогові дані телеметрії;

x_{ij} - цифрові дані телеметрії;

D_i – техніко-економічні дані;

$p_1, p_2 \ldots p_k$ – система взаємно простих модулів;

$b_1, b_2 \ldots\ldots b_j \ldots\ldots b_k$ – набір найменших невід'ємних залишків;

B_j – система ортогональних базисів СЗК.

9.3. Каскадне кодування даних на основі методу залишків та СЗК.

Процеси кодування та декодування інформації методом залишків виконуються наступним чином. Вхідний масив даних на деякому інтервалі часу описується матрицею

$$Y_{ij} = \begin{vmatrix} Y_{11} & \dots & Y_{i1} & \dots & Y_{k1} & \dots & Y_{n1} \\ Y_{12} & \dots & Y_{i2} & \dots & Y_{k2} & \dots & Y_{n2} \\ . & . & . & . & . & . & . \\ Y_{1k} & \dots & Y_{ik} & \dots & Y_{kk} & \dots & Y_{nk} \end{vmatrix}, \quad (9.11)$$

де Y_{ij}- цифровий відлік i-го вимірювання j-го каналу n, k- число відліків (n>k).

Застосувавши метод залишків, матрицю (9.11) перетворюємо у матрицю залишків $\|b_{ij}\|$ шляхом нелінійної модульної операції в коді поля Галуа.

$$\left\|b_{ij}\right\| = resY_{ij}\,(\mathrm{mod}P_j), i \in \overline{1,n};\ j \in \overline{1,k}, \quad (9.12)$$

де $P_j - 1 \geq 2\left|Y_{ij} - Y_{i-1j}\right|\max$ - є умовою однозначності перетворень.

Вираз (9.12)відповідає лінійній формі

$$Y_{ij} = a_{ij} \cdot P_j + b_{ij},$$

де $a_{ij} = \overset{\vee}{E}\left[\frac{Y_{ij}}{P_j}\right]$ - ранг відліку Y_{ij} по модулю P_j.

Тому для однозначного відновлення цифрових відліків матриці (9.11) на основі рекурентності формули

$$Y_{i+1,j} = \hat{E}\left[\frac{Y_{ij} - b_{i+1,j}}{P_j}\right]P_j + b_{ij} \quad (9.13)$$

необхідно матрицю залишків (9.12) доповнити вектором-строкою опорних рангів $a_{ij}(i \in \overline{1,k})$ і елементом X,

$$\begin{vmatrix} b_{11} & \dots & b_{i1} & \dots & b_{k1} & b_{k+1,1} & \dots & b_{n1} \\ b_{12} & \dots & b_{i2} & \dots & b_{k2} & b_{k+1,2} & \dots & b_{n2} \\ . & . & . & . & . & . & . & . \\ b_{1j} & \dots & b_{ij} & \dots & b_{kj} & b_{k+1,j} & \dots & b_{nj} \\ . & . & . & . & . & . & . & . \\ b_{1k} & \dots & b_{ik} & \dots & b_{kk} & b_{k+1,k} & \dots & b_{nk} \\ a_{1k+1} & \dots & a_{ik+1} & \dots & b_{k,k+1} & X_{k+1} & \dots & X \end{vmatrix}, \quad (9.14)$$

де елементи $0 \leq X \leq P_{k+1} - 1$ утворюють поле довільних інформаційних повідомлень по mod P_{k+1}.

Таким чином, однозначність перетворень матриць (9.14) та (9.11) забезпечується в розширеній СЗК по модулю P_{k+1}. Тоді, застосувавши

зворотнє перетворення СЗК у розширеній системі модулів $P = P_1 \cdot P_2 \cdot ... P_k \cdot P_{k+1}$, отримаємо

$$N_{i,k+1} == res \sum_{j=1}^{k+1} C_{ij} \cdot B'_j (\mathrm{mod} P),$$

де C_{ij} елементи векторів стовбців

$$C_{il} = \begin{pmatrix} b_{i1} \\ \dots \\ b_{ik} \\ a_{i,k+1} \end{pmatrix}, \qquad C_{il} = \begin{pmatrix} b_{i1} \\ \dots \\ b_{ik} \\ X \end{pmatrix}$$

$$1 \le i \le k; \qquad k+1 \le i \le n;$$

B_{ij} – ортогональні базиси розширеної СЗК.

Структура даного перетворення СЗК має наступний вигляд:

$$\left.\begin{array}{l} x_1(t) \to x_{i1} \to b_{i1}(\mathrm{mod} q_1) \to p_1 \to b_1 \\ x_2(t) \to x_{i2} \to b_{i2}(\mathrm{mod} q_2) \to p_2 \to b_2 \\ \dots \\ x_j(t) \to x_{ij} \to b_{ij}(\mathrm{mod} q_j) \to p_j \to b_j \\ \dots \\ x_{k-1}(t) \to x_{ik-1} \to b_{ik-1}(\mathrm{mod} q_{k-1}) \to p_{k-1} \to b_{k-1} \\ D_k \to p_k \to b_k \end{array}\right\} N_k = res \sum_{j=}^{k} b_j B_j (\mathrm{mod} P),$$

де D_k - службові алфавітно-цифрові дані

Для однозначності перетворення (9.15) діапазони квантування відліків A_j по каналах розраховуються по формулі

$$A_j = P_j \cdot P_{k+1} - 1.$$

Для захисту кодових слів $N_{i,k+1}$ від помилок необхідно введення деякої надлишковості шляхом застосування нерозділимих арифметичних AN-кодів, що дозволяє ефективно виявляти та виправляти помилки при відносно короткій їх двійковій довжині 16 - 64 біт.

Тому, розширивши набір модулів

$$\rho' = P_0 \cdot P_1 \cdot P_2 \cdot ... \cdot P_{k+1},$$

та визначивши новий набір базисних чисел $\{B_i''\}$ і доповнивши матрицю (9.14) нульовим стовбцем і строкою, отримаємо:

$$\begin{vmatrix} b_{0,0} & b_{1,0} & \dots & b_{k,0} & b_{k+1,0} & \dots & b_{n,0} \\ b_{1,0} & b_{1,1} & \dots & b_{k,1} & b_{k+1,1} & \dots & b_{n,1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{0,j} & b_{0,j} & \dots & b_{0,j} & b_{k+1,j} & \dots & b_{n,j} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{0,k} & b_{1,k} & \dots & b_{k,k} & b_{k+1,k} & \dots & b_{n,k} \\ X_0 & a_1 & \dots & a_k & X_{k+1} & \dots & X_n \end{vmatrix}. \qquad (9.15)$$

Описаний метод реалізований в інформаційній системі контролю процесів буріння (комплекси СКУБ та АТОС-Б), які серійно випускалися заводом «Промприлад» (м. Івано-Франківськ).

В названих системах вектор стовбець $b_{0,0},...b_{0,k}$ - використовується для ідентифікації блоку даних матриці залишків $b_{i,j}$ та стартової інформації. Символу X_0 присвоюється код стану об'єкта управління $X_0 = S_i$, параметри якого описуються блоком даних (9.15). Вектор-строка є нульовим, тобто всі значення $b_{i,0} = 0$, що забезпечує захист даних від помилок. Символи $X_{k+1},...X_n$ використані для кодування даних, які вводяться оператором з пульта і кодуються півбайтами (4 біти) по модулю $P_{k+1} = 16$.

Наприклад у процесорі СЗК терміналу АТОС-Б реалізований набір модулів для двохбайтових слів $N_{i,S}$ ($P_0 = 5, P_1 = 7, P_2 = 9, P_3 = 13, P_4 = 16$), добуток яких $\rho'' = 2^{16} - 5$, а діапазони квантування цифрових даних відповідно рівні:

$A_1 = 7 \cdot 16 - 1 = 111$, $\delta_1 < 1\%$

$A_2 = 9 \cdot 16 - 1 = 143$, $\delta_2 < 1\%$

$A_3 = 13 \cdot 16 - 1 = 207$, $\delta_3 < 0{,}5\%$,

де δ_i - похибка вимірювання технологічного параметра.

Застосування модуля $P_{k+1} = 16$ при кодуванні $N_{i,k+1}$ у двійковій системі числення дозволяє виділяти службову інформацію по $\bmod P_0$ без декодування чисел $N_{i,k+1}$ шляхом вилучення 4-х молодших бітів. Тобто:

$$b_{i,k} = (a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0)_N;\ a_i \in \overline{0,1}.$$

На рис.9.2 та 9.3 відповідно показана структура вихідного коду та зовнішній вигляд реалізованого в промисловості 32-канального процесора СЗК.

P_0	0	0	2	2	2	2	0	0	0	...	0	2	2	2	2	0	0	0	...
P_1	b_1	b_1	b_1	b_1	b_1	b_1	b_1	b_1	b_1	...	b_1	b_1	b_1	b_1	b_1	b_1	b_1	b_1	...
P_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2	...	b_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2	...
P_3	b_3	b_3	b_3	b_3	b_3	b_3	b_3	b_3	b_3	...	b_3	b_3	b_3	b_3	b_3	b_3	b_3	b_3	...
P_1	×	×	R_3	R_2	R_1	C_1	×	×	×	...	×	R_3	R_2	R_1	C_1	×	×	×	...
P_0	1	1	3	3	3	3	1	1	1	...	1	3	3	3	3	1	1	1	...
P_1	b_4	b_4	b_4	b_4	b_4	b_4	b_4	b_4	b_4	...	b_4	b_4	b_4	b_4	b_4	b_4	b_4	b_4	...
P_2	b_5	b_5	b_5	b_5	b_5	b_5	b_5	b_5	b_5	...	b_5	b_5	b_5	b_5	b_5	b_5	b_5	b_5	...
P_3	b_6	b_6	b_6	b_6	b_6	b_6	b_6	b_6	b_6	...	b_6	b_6	b_6	b_6	b_6	b_6	b_6	b_6	...
P_4	×	×	R_6	R_5	R_4	C_2	×	×	×	...	×	R_6	R_5	R_4	C_2	×	×	×	...

Рис.9.2 Структура вихідного коду процесора СЗК терміналу АТОС-Б.

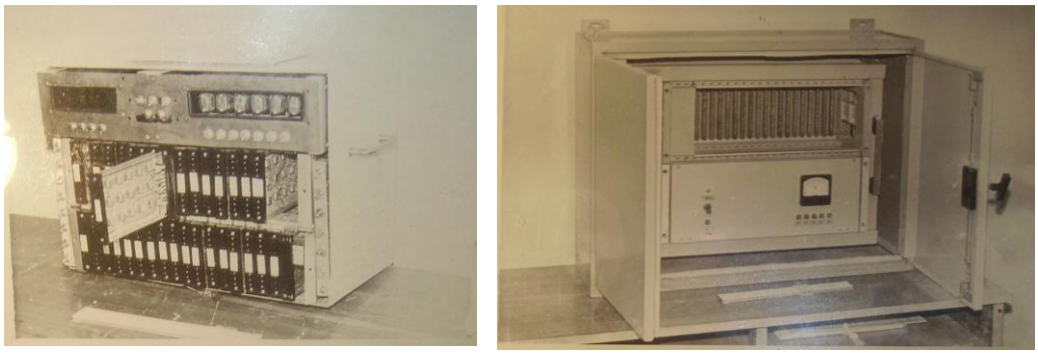


Рис.9.3 Зовнішній вигляд дослідного взірця 32-бітного процесора СЗК.

Позитивними якостями викладених методів пакування даних на основі сум полів Галуа та СЗК є наступні:

– строга зворотність та однозначність перетворення інформаційних повідомлень;

– неадаптивність процедур перетворення;

– рівномірність швидкості інформаційних потоків на виході кодера;

– формування байторієнтованих інформаційних даних;

– структуризованість блоків даних та їх захист від помилок;

– зменшення надлишковості інформації.

До недоліків слід віднести деяку громіздкість зворотного перетворення СЗК у цілочисельній формі згідно виразу (9.5), що включає операції множення, ділення та потребує розрахунку нормуючих вагових коефіцієнтів m_i.

9.4. Нормалізована форма СЗК.

Наявність операції згортки N_k по модулю P_j у виразі (9.6) при зростанні відношення ρ/P_j у великорозрядних процесорах знижує можливості створення високопродуктивних програмно-апаратних засобів опрацювання даних на основі теорії кодів поля Галуа. Аналогічно необхідність визначення залишку по модулю ρ у зворотному перетворенні цілочисельної форми СЗК від суми добутків $b_j \cdot B_j$ ускладнює реалізацію суматорів та знижує їх швидкодію. Цілочисельність подання даних також не завжди вигідна, оскільки числа в процесорах, як правило, кодуються у нормалізованій формі з плаваючою або фіксованою комою. Тому одним з шляхів оптимізації та підвищення швидкодії процесорів СЗК є приведення її перетворень до нормалізованої форми.

Умову нормування числа $N_{i,k}$ отримаємо розділивши ліву і праву частини виразу (9.5) на ρ

$$\frac{N_k}{\rho} = res\sum_{j=1}^{k} b_j \frac{\rho}{P_j} \cdot m_j (\mathrm{mod}\rho)/\rho .$$

У результаті отримаємо

$$[N_k]_0 = res\sum_{j=1}^{k} [b_j]_0 \cdot m_j (\mathrm{mod}\, 1), \qquad (9.16)$$

де $[N_k]_0 = N_k/\rho$; $[b_j]_0 = b_j/P_j$; операція mod1 - виконується шляхом відкидання цілої частини результату сумування добутків $[b_j] \cdot m_j$.

Враховуючи, що нормування по модулю P_j значення залишку b_j отримується при рішенні рівняння

$$[b_j]_0 = res[N_k]_0 \frac{\rho}{P_j} (\mathrm{mod}\, 1) \qquad (9.17)$$

відновлення цілочисельного значення b_j виконується згідно виразу

$$b_j = \hat{\mathrm{E}}[[b_j]_0 \cdot P_j], \qquad (9.18)$$

$\hat{\mathrm{E}}[\cdot]$ - цілочисельна функція з округленням до більшого цілого.

Наявність цілочисельної функції $\hat{\mathrm{E}}[\cdot]$ у виразі (9.17) обумовлена тим, що відношення b_j/P_j, як правило, є раціональним або безконечним

дробовим числом, яке задається скінченним числом розрядів з деякою від'ємною похибкою, яка викликає похибку обчислення b_j.

Для оцінки допустимої величини вказаної похибки по кожному модулю нормалізованої СЗК (НСЗК) запишемо відношення $b_j \diagup P_j$ у вигляді:

$$\frac{b_j}{P_j} = A_j + \delta_j, \tag{9.19}$$

де A_j - округлене цілочисельне значення відношення $b_j \diagup P_j$;

δ_j - похибка округлення.

Підставивши (9.19) у (9.16), отримаємо

$$[N_k]_0 = res(\sum_{j=1}^{k} b_j \cdot A_j + \sum_{j=1}^{k} b_j \cdot \delta_j) \bmod 1,$$

де сума, яка містить елементи δ_j, визначає загальну похибку обчислення $[N_k]_0$.

Згідно (9.17) N_k нормується по ρ з точністю до ρ^{-1}, тому однозначність перетворення інформації в НСЗК досягається, якщо величина похибки обчислення $[N_k]_0$ задовольняє умову

$$|\sum_{j=1}^{k-\nu} b_j \cdot \delta_j| < 0.5\rho;\ \nu \in \overline{1, k-1}. \tag{9.20}$$

Умова (9.20) дозволяє сформулювати задачу оцінки похибки округлення $b_j \diagup P_j$ у вигляді системи лінійних нерівностей. При цьому необхідно враховувати граничний випадок коли $b_j = P_j - 1$. З врахуванням останнього система вказаних нерівностей набуває вигляду:

$$\left.\begin{array}{c} |\rho(P_1-1)\delta_1| < 0{,}5 \\ |\rho(P_2-1)\delta_2| < 0{,}5 \\ \cdots \quad \cdots \quad \cdots \\ |\rho(P_k-1)\delta_k| < 0{,}5 \end{array}\right\}; \qquad 1)$$

$$\left.\begin{array}{c} |\rho[(P_1-1)\delta_1 + (P_2-1)\delta_2]| < 0{,}5 \\ |\rho[(P_1-1)\delta_1 + (P_3-1)\delta_3]| < 0{,}5 \\ \cdots \quad \cdots \quad \cdots \\ |\rho[(P_1-1)\delta_1 + (P_k-1)\delta_k]| < 0{,}5 \end{array}\right\}; \qquad 2)$$

$\cdots \qquad \cdots \qquad \cdots$

$$| \rho[(P_1 -1)\delta_1 + (P_2 -1)\delta_2 + ... + (P_k -1)\delta_k] | < 0{,}5\}. \qquad \text{k)}$$

Рішення отриманої системи нерівностей дозволяє визначити допустимі похибки подання парних добутків, при яких забезпечується однозначність перетворення НСЗК.

Враховуючи, що кількість рівнянь в розглянутій системі зростає (по закону 2^{k-1}), при збільшенні числа модулів кодування її рішення у загальному вигляді для різних наборів модулів порівняно з великим об'ємом обчислень.

Розбиваючи систему (9.21) на окремі групи: 1), 2),… k), можна оцінити величину допустимої похибки δ_j по кожному модулю, їх парах і т.д., використовуючи методи лінійної алгебри. Але, подання $[N_k]_0$ з похибкою, яка отримує як додатне, так і від'ємне значення, нераціонально з точки зору розшифрування інформації програмованими обчислювальними засобами. Наявність у системі команд сучасних процесорів операцій виділення цілої та дробової частин результатів обчислень, наприклад, цілочисельної функції з округленням до меншого цілого

$$A_j = \overset{\vee}{\mathrm{E}}[x_j];\ x_j = A_j + x_j;\ 0 \le \delta_j < 1,$$

а також велика швидкодія їх виконання визначає доцільність подання $[N_k]_0$ та $\lfloor b_j \rfloor_0$ тільки з додатною похибкою заокруглення. Це у свою чергу створює умови заміни процедур згортки по модулю ρ у перетворенні СЗК на процедури множення та виділення цілої частини результатів перетворення НСЗК.

У цьому випадку рішення системи (9.20) суттєво спрощується і зводиться до розв'язання однієї нерівності:

$$\rho[(P_1 -1)\delta_1 + (P_2 -1)\delta_2 + ... + (P_k -1)\delta_k] | < 1. \qquad (9.21)$$

При апаратурній або програмній реалізації перетворення НСЗК часто потрібна оцінка кількості необхідних двійкових розрядів відношення $b_j \big/ P_j$.

Таку оцінку знизу можна отримати, розв'язуючи першу групу системи нерівностей (9.21) відносно додатньої помилки округлення $\lfloor b_j \rfloor_0$ по кожному модулю P_j.

$$\delta_{j\max} < [\rho(P_j -1)]^{-1}. \qquad (9.22)$$

Перетворивши (9.22) шляхом логарифмування за основою 2 отримаємо:

$$\delta_{j\max} < 2^{-\hat{\mathrm{E}}[\log_2 \rho(P_j -1)]}, \qquad (9.23)$$

де $\hat{\mathrm{E}}[\cdot]$ - необхідна кількість двійкових розрядів дробового числа b_j/P_j округленого в сторону більшого.

На практиці, при створенні процесорів, які реалізують цифрові перетворення розрядність подання даних по кожному модулю P_j вибирається однаковою. Тоді $\delta_1 = \delta_2 = ... = \delta_k$, і рішення нерівності (9.21) отримує вигляд:

$$\delta_{доп} < [\rho \sum_{j=1}^{k} (\mathrm{P_j} - 1)],$$

або після логарифмування

$$\delta_{доп} < 2^{-\hat{\mathrm{E}}[\rho \sum_{j=1}^{k} (P_j - 1)]}, \qquad (9.24)$$

$\delta_{доп}$ - допустима системна похибка округлення по кожному з модулів P_j.

У відповідності з вищевикладеним пряме та зворотнє перетворення НСЗК можна записати у вигляді

$$b_j = \check{\mathrm{E}}\left[res\left[N_k\right]_0 \cdot \overset{\circ}{\mathrm{B}}_j (\mathrm{mod}\, 1) \cdot P_j \right];$$

$$\left[N_k\right]_0 = res \sum \left[b_j\right]_0 \cdot m_j (\mathrm{mod} 1),$$

де $$\overset{\circ}{\mathrm{B}}_j = \begin{cases} \rho/P_j, якщо\_m_j = 1 \\ -\rho/P_j, якщо\_m_j = P_j - 1. \end{cases}$$

В останніх виразах операція згортки по модулю одиниця еквівалентна виділенню дробової частини результату обчислень, $\check{\mathrm{E}}[\cdot]$ - відповідно виділенню цілої частини, а від'ємне значення базисного числа $\overset{\circ}{\mathrm{B}}_j$ відповідає реалізації зворотнього перетворення НСЗК на основі найменших залишків, які можуть приймати від'ємні значення. Тобто $-P_j/2 \le \left[\overset{\circ}{b}_j\right] \le P_j/2$.

Так як величина $\delta_{доп}$ задовольняє будь якому відношенню b_j/P_j, $a \le b_j < P_j - 1,\ 1 \le m_j < P_j - 1$, то алгоритм прямого та зротнього перетворень НСЗК отримують вид:

$$[b_k]_P = res[N_k]_\rho \cdot \overset{\circ}{\mathrm{B}}_j (\mathrm{mod}\, 1);$$

$$[N_k]_\rho = res \sum_{j=1}^{k} [b_j]_P \cdot m_j (\mathrm{mod} 1);$$

або у вигляді

$$[N_k]_\rho = \ldots \sum_{j=1}^{k} [b_j]_P \cdot m_j;$$

$$[b_k]_P = \ldots {b_j}/{P_j}; \qquad (9.25)$$

$$[b_k]_P = \ldots [N_k]_\rho \cdot \overset{\circ}{\mathrm{B}}_j.$$

Структура перетворення СЗК в нормалізованій формі має наступний вигляд:

$$\left.\begin{array}{l} x_1(t) \to x_{i1} \to p_1 \to [b_1]_0 \\ x_2(t) \to x_{i2} \to p_2 \to [b_2]_0 \\ \ldots \\ x_j(t) \to x_{ij} \to p_j \to [b_j]_0 \\ \ldots \\ x_m(t) \to x_{im} \to p_{k-1} \to [b_{k-1}]_0 \\ D_i \to p_k \to [b_k]_0 \end{array}\right\} [N_k]_0 = res \sum_{i=1}^{k} [b_i]_0 (\mathrm{mod} 1); \quad (9.26)$$

$$b_i = \mathrm{int}\, res[N_k]_0 (\mathrm{mod}\, 1) \cdot P,$$

де int - символ виділення цілої частини результату обчислень.

Позитивною якістю отриманих виразів НСЗК є подання $[N_k]_0$ та $\lfloor b_j \rfloor_0$, при виконанні умови (9.24), у формі нормалізованих двійкових чисел з фіксованою комою, що забезпечує високу степінь сумісності кодування даних та виконання перетворень НСЗК на універсальних та спеціалізованих процесорах.

Розглянемо приклади.

Нехай задано систему модулів НСЗК $P_1 = 3, P_2 = 5, P_3 = 7$, для якої $\rho = 105;\ m_1 = 2;\ m_2 = 1;\ m_3 = 1$.

Визначимо допустиму похибку округлення перетворень НСЗК

$$\delta_{доп} = \rho^{-1} = {1}/{105} = 0{,}00952_{(10)} = 0{,}00000010_{(2)}$$

Обчислимо таблиці нормалізованих значень $\lfloor b_j \rfloor_p$

а) у десятковій системі числення:

b_j	$P_1 = 3$	$P_2 = 5$	$P_3 = 7$
0	0,00	0,00	0,00
1	0,33	0,20	0,143
2	0,66	0,40	0,286
3	-	0,60	0,428
4	-	0,80	0,571
5	-	-	0,714
6	-	-	0,857

б) у двійковій системі числення:

b_j	$P_1 = 3$	$P_2 = 5$	$P_3 = 7$
0	0,00000000	0,00000000	0,00000000
1	0,01010101	0,00110011	0,00011111
2	0,10101011	0,01100110	0,00111110
3	-	0,10011001	0,01101101
4	-	0,11001100	0,01111101
5	-	-	0,01010111
6	-	-	0,11011011

Задано $N_k = 1$, тоді $b_1 = b_2 = b_3 = 1$ і $[N_k]_0 = 0{,}00000010_{(2)}$.

Виконаємо зворотнє перетворення НСЗК згідно виразу (9.25)

$$[N_k] = (0{,}33 \cdot 2 + 0{,}2 \cdot 1 + 0{,}143 \cdot 1) \bmod 1 = 0{,}00966$$

звідки

$$N_k = \overset{\vee}{\mathrm{E}}[\rho \cdot [N_k]_0] = \overset{\vee}{\mathrm{E}}[105 \cdot 0{,}00966] = 1.$$

Аналогічно виконуємо перетворення НСЗК у двійковій системі числення, тобто:

$$[b_1]_0 = 0{,}01010101_{(2)};$$
$$[b_2]_0 = 0{,}00110011_{(2)};$$
$$[b_3]_0 = 0{,}00100100_{(2)},$$

звідки

$$[N_k]_0 = (2\cdot[b_1]_0 + 1\cdot[b_2]_0 + 1\cdot[b_3]_0)\bmod 1 =$$

$$\begin{array}{r} 0{,}10101010 \\ +\ 0{,}00110011 = 0{,}00000011_{(2)} \\ +\ \underline{0{,}00100100} \\ 1{,}00000011 \end{array}$$

Отже

$$N_k = \overset{\vee}{\mathrm{E}}[\rho\cdot[N_k]_0] = \overset{\vee}{\mathrm{E}}[105_{(10)}\cdot[N_k]_0]$$

Оскільки $N_k = 105_{(10)} = 1101001_{(2)}$, отримаємо N_k перемноживши його у двійковій системі числення на $[N_k]_0$

$$\begin{array}{r} 001101001 \\ \times\ \underline{0{,}00000010}, \\ 001{,}101010 \end{array}$$

що відповідає дійсному значенню $N_k = \overset{\vee}{\mathrm{E}}[1{,}101010] = 1$.

Допустимо: $b_1 = 2$; $b_2 = 4$; $b_{31} = 6$, тобто

$$N_k = (2,\ \ 4,\ \ 6) = \rho - 1 = 104_{(10)}.$$

Виконаємо пряме і зворотнє перетворення НСЗК наступним чином:

$b_1 = 2/3 = 0{,}66_{(10)} = 0{,}10101010_{(2)}$;

$b_2 = 4/5 = 0{,}8_{(10)} = 0{,}11001100_{(2)}$;

$b_3 = 6/7 = 0{,}85_{(10)} = 0{,}11011011_{(2)}$,

$$[N_k]_0 = (2\cdot 0{,}10101010 + 1\cdot 0{,}11001100 + 1\cdot 0{,}11011011)\bmod 1 =$$

$$= \begin{array}{r} 0{,}01010101 \\ +\ 0{,}11001100 \\ +\ \underline{0{,}11011011} \\ 1{,}11110100 \end{array} = 0{,}11110100_{(2)}.$$

Обчислимо N_k

$$N_k = \overset{\vee}{\mathrm{E}}[105_{(10)}\cdot[N_k]_0] = 1100100_{(2)}.$$

Виконаємо перевірку обчислень шляхом множення двійкових кодів:

$$
\begin{array}{rl}
 & 0{,}11110100 \\
\times & \quad 1101001 \\
\hline
 & \quad 11110100 \\
+ & \quad 11110100 \\
 & \quad 11110100 \\
 & \quad 11110100 \\
\hline
 & 1100100{,}00010100
\end{array}
$$

$1100100_{(2)} = 104_{(10)} = N_k = \rho - 1$.

Таким чином однозначність перетворень НСЗК доведена.

9.5. Досконалі форми СЗК.

Основною операцією, яка збільшує часову складність алгоритмів перетворень в цілочисельній та нормалізованій СЗК, є процедура множення при отриманні парних добутків $b_j \cdot B_j$ та $[b_j] \cdot m_j$. Тому її виключення з аналітики названих перетворень дозволяє добитися суттєвих спрощень, як в алгоритмічному, так і в апаратурному аспектах.

Для рішення цієї задачі запишемо вираз для зворотнього перетворення СЗК з одиничними ваговими коефіцієнтами

$$
N_k = \begin{cases} res \sum_{j=1}^{k} b_j (\pm 1) \frac{\rho}{P_j} (\mathrm{mod}\, \rho), & \dot{N}_k \le 0; \\ \rho - res \sum_{j=1}^{k} b_j (\pm 1) \frac{\rho}{P_j} (\mathrm{mod}\, \rho), & \dot{N}_k > 0. \end{cases} \tag{9.27}
$$

Зворотнє перетворення для таких СЗК у випадку (9.27) визначається виразом

$$
\pm \dot{N}_k = res \sum_{j=1}^{k} b_j (\pm 1) \frac{\rho}{P_j} (\mathrm{mod}\, \rho), \tag{9.28}
$$

а пряме пертворення виконується згідно виразу

$$
b_j = \begin{cases} res \dot{N}_k (\mathrm{mod} P_j), & N_k \le 0; \\ res(\rho - \dot{N}_k)(\mathrm{mod} P_j), & N_k > 0. \end{cases}
$$

Умови (9.27) та (9.28) дозволяють сформувати вимоги до наборів модулів СЗК, в яких виключається надлишковість подання $b_j \cdot B_j$ при заданих $\{m_j\}$, тобто:

$$
\begin{aligned}
&0)\ \{m_j\} = \{1,1,\ldots,1\}; \quad \{P_j > m_j\};\\
&1)\ \{m_j\} = \{1,1,\ldots,P_k - 1\}; \quad P_k = m_k + 1;\\
&2)\ \{m_j\} = \{1,1,\ldots,P_{k-1} - 1, P_k - 1\}; \quad P_{k-1} = m_{k-1} + 1; P_k = m_k + 1;\\
&\ldots \qquad\qquad \ldots\\
&k)\ \{m_j\} = \{P_j - 1\}; \quad P_j = m_j + 1.
\end{aligned}
\tag{9.29}
$$

Розглянемо випадок (9.29.0).

Підставивши $m_j = 1$ у вираз для B_j (9.16) визначимо умову існування СЗК з набором m_j (9.29.0), перейшовши від порівняння до лінійного рівняння

$$\sum_{j=1}^{k} \prod_{i=j}^{k} P_i = r_1 \prod_{j=1}^{k} P_j + 1, \tag{9.30}$$

де $r_1 = 0,1,2,...$ - ранг СЗК.

При зміні параметра $k = 1,2,\ldots$ згідно останнього виразу маємо:

$k = 1; \quad P_1 = r_1 P_1 + 1;$

$k = 2; \quad P_1 + P_2 = r_1 P_1 \cdot P_2 + 1;$

$k = 3; \quad P_1 \cdot P_2 + P_1 \cdot P_3 + P_2 \cdot P_3 = r_1 P_1 P_2 P_3 + 1;$

… …

Очевидно, що при $k = 1$ і $k = 2$ не може бути знайдено ні однієї СЗК з властивістю (9.29.0). В інших випадках (9.30) доцільно привести до вигляду

$$x_k \sum_{j=1}^{k-1} \prod_{i \neq j}^{k} P_i + \prod_{i \neq k}^{k} P_i = r_1 x_k \prod_{j \neq k}^{k} P_j + 1,$$

або кінцево

$$x_k = \frac{\prod_{i \neq k}^{k} P_i - 1}{r_1 \prod_{j \neq k}^{k} P_j - \sum_{j=1}^{k} \prod_{i \neq j}^{k} P_i}, \tag{9.31}$$

де $x_k = P_k$-й модуль СЗК, який задовольняє умові (9.30) і обчислюється на основі відомих $k-1$ модулів.

Наприклад:

Нехай заданий набір модулів СЗК

$$\{P_j\} = \{2,3,7,x\}.$$

Необхідно знайти таке x, щоб $\{m_j\} = \{1,1,1,1\}$. Підставляючи числові значення цього прикладу в (9.31) отримаємо

$$x = \frac{42-1}{42-(6+14+21)} = \frac{41}{42-41}$$

і при $r_1 = 1,\ x = 41.$

Так як отримане значення x просте число, рішення є єдиним.

Таким чином отримуємо набір модулів досконалої СЗК

$$\{P_j\} = \{2,3,7,41\}, \quad \rho = 1722$$

Для перевірки правильності рішення (9.31) розрахуємо ортогональні базиси СЗК з таким набором модулів

$$\dot{B}_1 = \frac{1722}{2} m_1 \equiv 1(\mathrm{mod}2);\ m_1 = 1;$$

$$\dot{B}_2 = \frac{1722}{3} m_2 \equiv 1(\mathrm{mod}3);\ m_2 = 1;$$

$$\dot{B}_3 = \frac{1722}{7} m_3 \equiv 1(\mathrm{mod}7);\ m_3 = 1;$$

$$\dot{B}_4 = \frac{1722}{41} m_1 \equiv 1(\mathrm{mod}41);\ m_4 = 1.$$

Отже

$$\dot{N}_k = (\dot{B}_1 + \dot{B}_2 + \dot{B}_3 + \dot{B}_4)(\mathrm{mod}\ \rho) = 1,$$

Тобто для нашого прикладу

$$\dot{N}_k = (861 + 574 + 246 + 42)(\mathrm{mod}\ 1722) = 1,$$

що відповідає умові $\{m_j\} = \{1,1,1,1\}$.

На рис.9.4 графічно показані рішення (9.31), розраховані для різних наборів модулів $\{P_j\}$ та значень r_1 і k. Для виразності ілюстрації вказані графіки подані в логарифмічному масштабі.

На рисунку показано, що для $k = 3$ існує єдиний набір модулів $\{2,3,5\}$, якому відповідає набір $\{m_j\} = \{1,1,1,1\}$. Зі збільшенням k кількість такого класу СЗК зростає.

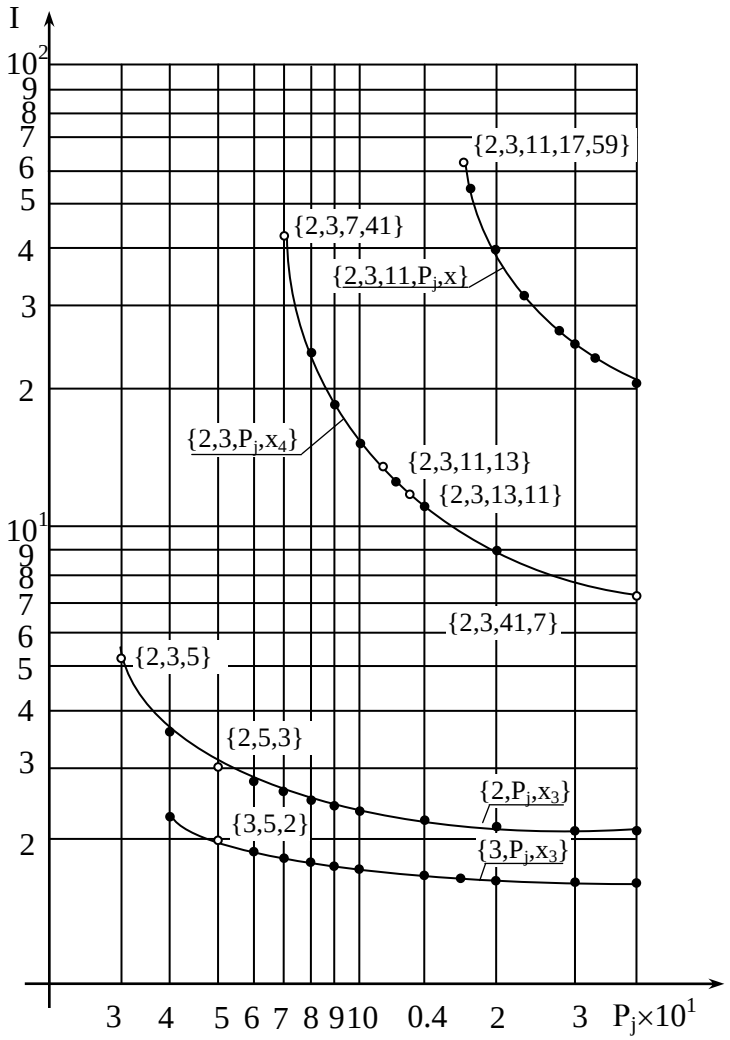


Рис. 9.4. Набори модулів досконалої форми СЗК.

Аналогічно знайдемо вирази існування СЗК для умов (9.29.1-к):

$$
\begin{aligned}
&1)\ \sum_{j=1}^{k-1}\prod_{i\neq j}^{k-1} P_i - \prod_{j\neq k}^{k} P_i = \pm r_1 \prod_{j=1}^{k} P_j + 1;\\
&2)\ \sum_{j=1}^{k-2}\prod_{i\neq j}^{k-2} P_i - \sum_{j=k-1}^{k}\prod_{i\neq j}^{k} P_i = \pm r_1 \prod_{j=1}^{k} P_j + 1;\\
&\quad\ldots \qquad\qquad \ldots\\
&k)\ \sum_{j=1}^{k-2}\prod_{i\neq j}^{k-2} P_i - \sum_{j=1}^{k}\prod_{i\neq j}^{k} P_i = \pm r_1 \prod_{j=1}^{k} P_j + 1.
\end{aligned}
\qquad (9.32)
$$

Дослідження отриманих виразів показує, що умова (9.32.к) повністю співпадає з (9.30) Є, а також відображає параметри існування СЗК з одиничними ваговими коефіцієнтами $\{m_j\} = \{-1,-1,...,-1\}$.

Приведення СЗК досконалої форми дозволяє повністю виключити модульні операції у його зворотньому перетворенні.

Умову утворення досконалої без рангової СЗК можна отримати у вигляді нерівності, підставивши в (9.16) граничні значення $b_j = m_j = P_{j-1}$, тобто

$$\sum_{j=1}^{k}(P_j - 1)\frac{(P_j - 1)\cdot\rho}{P_j}) < \prod_{j=1}^{k} P_j, \qquad (9.33)$$

Звідси після елементарних перетворень отримаємо

$$\sum_{j=1}^{k}(P_j - 2 + \frac{1}{P_j}) < 1.$$

Очевидно, що у цьому випадку не існує ні одного набору цілочисельних взаємопростих модулів $P_j \geq 2$, $k \geq 2$, які утворюють досконалу без рангову СЗК.

Враховуючи, що коли $m_j = P_j - 1$,Є то згідно (9.29.к) можна замінити $m_j = -1$ і направленість (9.33) замисати у вигляді

$$\sum_{j=1}^{k}(P_j - 1)\frac{\rho}{P_j} > -\prod_{j=1}^{k} P_j$$

або

$$\sum_{j=1}^{n}(1 - \frac{1}{P_j}) < 1$$

Звідки згідно (9.33) при $k \geq 2$ та коли не існує ні однієї без рангової СЗК.

В якості одного з найбільш простих способів представлення СЗК до досконалої безрангової форми можна застосувати метод надлишкового розширення системи модулів початкової СЗК.

Умова існування досконалої без рангової СЗК для розширеної системи модулів визначається згідно виразу

$$\sum_{j=1}^{k-e}\prod_{i\neq j}^{k-e} P_i - \sum_{j=e}^{k-\nu} P_i \lhd \pm\prod_{j=1}^{k} P_j,$$

де e, ν - відповідно кількість основних і надлишкових модулів.

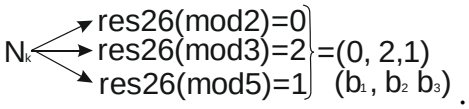
Розглянемо приклад:

Нехай маємо набір модулів досконалої СЗК

$$\{P_j\} = \{2, 3, 5\}; \quad \rho = 30; \quad \overset{0}{B}_1 = 15; \quad \overset{0}{B}_2 = 10; \quad \overset{0}{B}_3 = 6;$$

$m_1 = m_2 = m_3 = 1; \quad N_k = 26_{(10)} = 11010_{(2)}$.

Представимо число N_k у СЗК, тобто:



Запишемо залишки b_1, b_2, b_3, у нормалізованій формі:

$$[b_1]_0 = 0/2 = 0; \quad [b_2]_0 = 2/3 = 0{,}66; \quad [b_3]_0 = 1/5 = 0{,}2 .$$

Виконуємо зворотнє перетворення досконалої нормалізованої форми СЗК згідно виразу

$$[N_k]_0 = (0 + 0{,}66 + 0{,}2) \bmod 1 = 0{,}86 .$$

Отримаємо цілочисельне значення $[N_k]_0$

$$N_k = \hat{E}[[N_k]_0 \cdot \rho] = \hat{E}[0{,}86 \cdot 30] = 26 .$$

Аналогічно у двійковій системі числення виконаємо операції перетворень ДНСЗК:

$$[b_1]_0 = 0{,}00000_{(2)}; \quad [b_2]_0 = 0{,}10101_{(2)}; \quad [b_3]_0 = 0{,}00110_{(2)};$$

$$[N_k] = \begin{array}{r} 0{,}00000 \\ 0{,}10101 \\ 0{,}00110 \\ \hline 0{,}11011 \end{array}$$

$$\begin{array}{r} 0{,}11011 \\ \times\ 11110 \\ \hline 110110 \\ 11011 \\ 11011 \\ 11011 \\ \hline \end{array}$$

$N_к = \hat{E}[1100101010] = 26_{10}$

Труднощі пошуку досконалої форми СЗК, а також їх відносна рідкість вимагають розвитку інших методів отримання досконалих перетворень базису Крестенсона-Галуа.

Одним з таких методів є отримання скороченої СЗК з ваговими коефіцієнтами $\{m_j\} = \{1, 1, ..., 1\}$ шляхом виключення одного або кількох модулів у перетвореннях СЗК.

Умова існування таких СЗК має вигляд

$$\sum_{j=1}^{k-\nu} \prod_{i\neq 1}^{k-\nu} P_i = \prod_{j=1}^{k} P_j - 1 - \sum_{j-\nu}^{k} \prod_{i\neq j}^{k} P_i$$

Наприклад виберемо в якості початкового набору модулів СЗК наступні

$$\{P_j\} = \{5, 7, 8, 9, 19\},$$

для якого $\{m_j\}=\{1,1,1,1,8\}$, і виключимо з нього модуль $P_5=19$ прийнявши $b_5=0$.

Тоді всі операції перетворення в СЗК можуть виконуватись по алгоритмах безрангових досконалих форм СЗК.

Іншим універсальним практичним методом приведення до досконалої форми СЗК для будь яких наборів модулів $\{P_j\}$ є використання властивості (9.34).

Суть такого методу приведення в СЗК до досконалої форми полягає в тому, що при виконанні зворотнього перетворення СЗК апріорно задається набір $\{m_j\}=\{1,1,...,1\}$ для всіх модулів $\{P_j\}$. При цьому легко показати, що однозначність перетворення СЗК не порушується, але при виконанні зворотного перетворення необхідно коректувати значення добутих N_k залишків b_j з врахуванням конкретних значень $\{m_1, m_2, ..., m_j\}$ для заданого набору $\{P_1, P_2, ..., P_j\}$ згідно виразу:

$$b_j = res\, \overset{0}{b}_j \cdot \bmod P_j,$$

де $\overset{0}{b}_j$ - залишок отримання із числа $\overset{0}{N}_k$, тобто

$$\overset{0}{b}_j = res\, N_k (\bmod\ \overset{0}{P}_j),$$

а число $\overset{0}{N}_k$ досконалого перетворення СЗК розраховується згідно виразу

$$\overset{0}{N}_k = \left(\overset{0}{B}_1 b_1 + \overset{0}{B}_2 b_2 + ... + \overset{0}{B}_k b_k \right) (\bmod \rho),$$

де $\overset{0}{B}_0 = \rho / P_j \cdot 1$. Тобто: всі $m_j = 1; \quad j \in \overline{1,k}.$

Нехай заданий набір модулів СЗК: $\{P_j\}=\{P_1, P_2, \ldots, P_k\}, \quad \rho = \prod_{j=1}^{k} P_j,$

якому відповідають набори

$$\{m_j\}=\{m_1, m_2, ..., m_k\}, \quad \{B_j\}=\left\{\frac{\rho}{P_j}\cdot m_j\right\},$$

причому, не виконується ні одна з умов (9.35) і ні одним з розглянутих раніше методів СЗК з таким набором модулів не може бути приведене до досконалої форми.

Тоді, якщо прийняти всі $\left\{\overset{0}{m}_j\right\}=\{1,1,\ldots,1\}$ незалежно від їх фактичних значень $\{m_j\}$, то однозначність перетворень СЗК не порушується при виконанні їх згідно виразів

$$b_j = res\left[res\ \overset{0}{N}_k \left(\mathrm{mod}\, P_j\right)\cdot m_j\right]\mathrm{mod}\, P_j.$$

$$\overset{0}{N}_k = res\left(\overset{0}{B}_1\cdot b_j + \overset{0}{B}_2\cdot b_2 + \ldots + \overset{0}{B}_k\cdot b_k\right)\mathrm{mod}\ \rho$$

В нормалізованій досконалій формі СЗК аналогічні перетворення виконуються згідно виразів:

$$\left[\overset{0}{b}_j\right]_0 = b_j / P_j;\quad \left[\overset{0}{N}_k\right]_0 = res\sum_{j=1}^{k}\left[\overset{0}{b}_j\right]_0 (\mathrm{mod}1);$$

$$\overset{0}{N}_k = \hat{E}\left[\left[\overset{0}{N}_k\right]_0\cdot\rho\right];$$

$$b_j = \overset{\vee}{E}\left[\left[\overset{0}{b}_j\right]\cdot resP_j\cdot m_j\left(\mathrm{mod}\, P_j\right)\right].$$

Операція обчислення залишків $\overset{0}{b}_j$ з врахуванням $\{m_j\}$ виконується шляхом циклічного зсуву на величину нормалізованих значень з модулів P_j.

Наприклад:

Нехай маємо набір модулів СЗК:

P_1=5; P_2=7; P_3=11; тоді $\rho = 5\cdot 11\cdot 13 = 385;$

$B_1 = 77\cdot m_1 1(\mathrm{mod}5);\quad m_1 = 3;\quad B_1 = 231;\quad \overset{0}{B}_1 = 77;$

$B_2 = 55\cdot m_2 1(\mathrm{mod}7);\quad m_2 = 6;\quad B_2 = 230;\quad \overset{0}{B}_2 = 55;$

$B_3 = 35\cdot m_3 1(\mathrm{mod}11);\quad m_3 = 6;\quad B_3 = 210;\quad \overset{0}{B}_3 = 35.$

Таблиці значень залишків b_j для заданої (а) та досконалої форми СЗК (б) зваженої цілочисельної форми СЗК мають наступний вигляд:

b_i	P_1	P_2	P_3
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	–	5	5
6	–	6	6
7	–	–	7
8	–	–	8
9	–	–	9
10	–	–	10

а)

$\Rightarrow$

$\overset{o}{b}_i$	P_1	P_2	P_3
0	0	0	0
1	3	6	6
2	4	1	7
3	0	2	8
4	1	2	9
5	–	4	10
6	–	5	1
7	–	–	2
8	–	–	3
9	–	–	4
10	–	–	5

б)

Таким чином у цілочисельній формі СЗК число N_k=1 обчислюється згідно виразу

$$N_k = (231 \cdot 1 + 330 \cdot 1 + 210 \cdot 1)(\text{mod } 385) = 1$$

або

$$\overset{0}{N}_k = (77 \cdot 3 + 55 \cdot 6 + 35 \cdot 6)(\text{mod } 385) = 1,$$

Аналогічно складаються таблиці нормалізованих значень $\left[\overset{0}{b}_j\right]_0$ у десятковій та двійковій системах числення.

9.6. Міжбазисні перетворення на основі розмежованої СЗК.

Теоретичні основи та принципи реалізації розмежованої СЗК викладені в опублікованих одноосібних та сумісних з автором наукових працях О.І. Волинського, де викладені методи між базисних перетворень Радемахера-Крестенсона та Крестенсона-Радемахера.

Для створення повної функціонально-ефективної розмежованої СЗК (РСЗК) потрібно спростити існуючі алгоритми виконання операцій порівняння чисел. Умову створення швидкісного пристрою в РСЗК задовольняють алгоритми обчислення всіх арифметичних операцій крім ділення. Для реалізації операції ділення потрібна НДСЗК. Теоретичною основою даної форми є рівняння (9.16), підставивши в яке $m_1=m_2=m_k=1$, отримаєм базове рівняння перетворення НДСЗК у вигляді:

$$[N_k]_0 = res\sum_{i=1}^{k}[b_i]_0 (\text{mod } 1). \qquad (9.34)$$

З рівняння (9.34) видно, що з перетворення НДСЗК виключена операція множення і саме перетворення виконується у вигляді сумування нормалізованих залишків $[b_i]_0$ по mod1, що відповідає операції відкидання цілої частини результату.

Розглянемо приклад застосування НДСЗК.

Таблиця 9.1.

Перетворення векторів багатомірного дискретного простору залишків у одномірний дискретний простір Радемахера.

N_k	$[b_1]_0$	$[b_2]_0$	$[b_3]_0$	$[N_k]_0$
0	0	0	0	0
1	0.5	0.33	0.2	0.03
2	0	0.66	0.4	0.06
3	0.5	0	0.6	0.1
4	0	0.33	0.8	0.13
5	0.5	0.66	0	0.16
6	0	0	0.2	0.2
7	0.5	0.33	0.4	0.23
…	…	…	…	…
29	0.5	0.66	0.8	0.96

На основі табл. 9.1 запропонований спрощений алгоритм перетворення чисел з базису Крестенсона в базис Радемахера, що суттєво упростило здійснення операції порівняння, яка є базовою при здійсненні операцій віднімання та ділення.

Розглянемо приклад порівняння двох чисел в НДСЗК. Нехай: $N_{k1}=5$, $N_{k2}=7$.

Дані числа представлені згідно табл. 9.1 у вигляді: N_{k1}= (0,5;0,66;0) і N_{k2}=(0,5;0,33;0,4) і можуть бути нормалізовані у вигляді: $[N_{k1}]_0$= 0,16 і $[N_{k2}]_0$=0,23.

Графічно операцію вибірки залишків з табл. 9.1 можна представити у формі графа (рис.9.5).

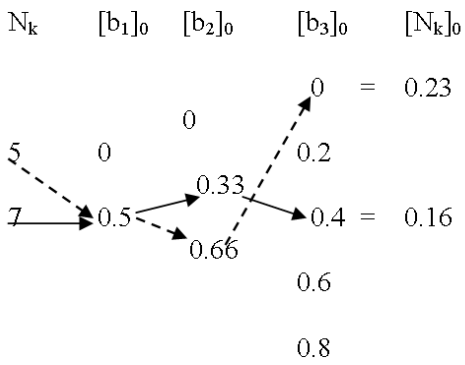


Рис. 9.5. Граф перетворення чисел з НДСЗК у базис Радемахера.

Витративши ще один такт на сумування залишків, отримаємо нормалізоване число, з яким потрібно зробити однотактну операцію віднімання. Далі порівнюємо результат з нулем і визначаємо знак результату, який необхідний для реалізації операції ділення в НДСЗК: $[N_{k1}]_0 - [N_{k2}]_0 \gtrless 0$.

На рис.9.6 показана структура спецпроцесора порівняння чисел, представлених у базисі Крестенсона на основі НДСЗК.

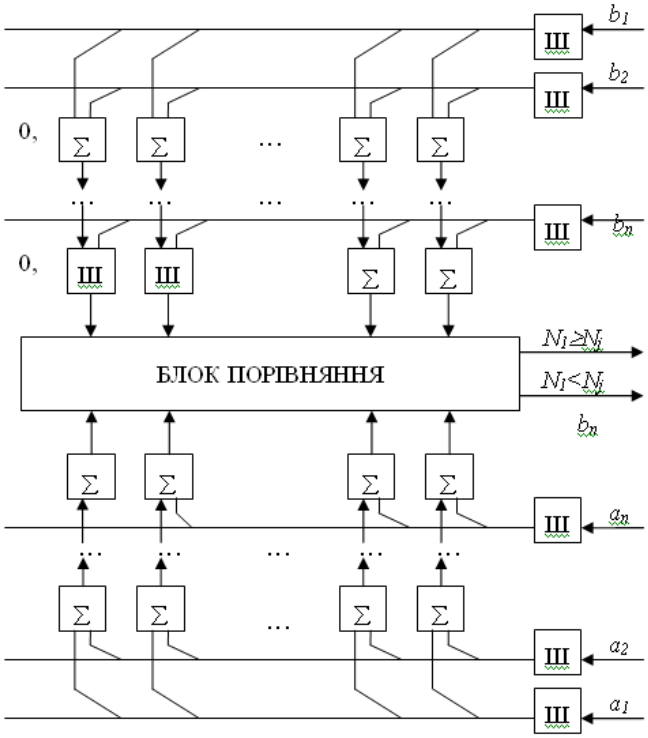


Рис.9.6. Структура спецпроцесора порівняння чисел у базисі Крестенсона.

9.6.1. Розрахунок набору модулів для реалізації 16-бітного процесора.

У таблиці 9.1 приведений приклад набору модулів, які відповідають досконалій формі СЗК. Ця умова обмежує можливості реалізації процесорів певної розрядності з оптимальним набором модулів. Оскільки довільному набору взаємопростих модулів відповідає довільний набір коефіцієнтів m_i.

Запропонований алгоритм приведення будь-якої цілочисельної форми СЗК до НДСЗК, який реалізується шляхом корекції графа визначення залишків згідно розрахованого набору коефіцієнтів m_i.

Для реалізації 16-бітного процесора оптимальний набір модулів з максимальним діапазоном кодування задається масивом чисел:

p_i=(3, 5, 7, 8, 11, 13)

Виконаємо розрахунок параметрів даної СЗК.

Діапазон кодування: $P = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 120120$

Базисні числа:

$$B_1 = \frac{120120}{3} \cdot m1 = 1(\mathrm{mod}3), m1 = 2; B_1 = 80080;$$

$$B_2 = \frac{120120}{5} \cdot m2 = 1(\mathrm{mod}5), m2 = 4; B_2 = 96096;$$

$$B_3 = \frac{120120}{7} \cdot m3 = 1(\mathrm{mod}7), m3 = 5; B_3 = 85800;$$

$$B_4 = \frac{120120}{8} \cdot m4 = 1(\mathrm{mod}8), m4 = 7; B_4 = 105105;$$

$$B_5 = \frac{120120}{11} \cdot m5 = 1(\mathrm{mod}11), m5 = 7; B_5 = 76440;$$

$$B_6 = \frac{120120}{13} \cdot m6 = 1(\mathrm{mod}13), m6 = 4; B_6 = 36960.$$

Скоректований граф обчислення залишків згідно розрахованого набору коефіцієнтів m_i буде мати вигляд:

$B_1 = 3$		$B_2 = 5$		$B_3 = 7$		$B_4 = 8$		$B_5 = 11$		$B_6 = 13$
0		0		0		0		0		0
0,6666	+	0,8	+	0,7142857	+	0,875	+	0,6363636	+	0,3076923 $\equiv (\mathrm{mod}\,1)$ 0,0000083
0,3333		0,6		0,428571		0,75		0,272727		0,615384
		0,4		0,142857		0,625		0,909091		0,923076
		0,2		0,8571428		0,5		0,545454		0,230769
				0,571428		0,375		0,181818		0,538461
				0,285714		0,25		0,818181		0,846153
						0,125		0,454545		0,153846
								0,090909		0,461538
								0,727272		0,769230
								0,363636		0,076923
										0,384615
										0,692307

З схеми обчислення числа за графом зрозуміло, що це значно спрощує перехід в базис Радемахера і виконання операції порівняння чисел. Використання такої схеми обчислення залишків дозволить побудувати арифметично овнофункціональний супершвидкісний 16-розрядний процесор.

9.6.2. Інформаційна база розмежованої СЗК.

Теоретичною основою РСЗК є цілочисельна форма СЗК, рівняння якої представлено у вигляді суми:

$$N_k = N_{1k} + N_{2k} + ... + N_{ik} + ... + N_{nk},$$

де N_{ik} – m-розрядний (розмежований) фрагмент числа N_k, яке представлене у двійковій системі числення, числового базису Радемахера. Наприклад 32-х розрядний процесор СЗК може бути розмежований на 4 фрагменти по 8 біт (рис.9.7).

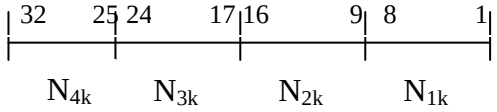
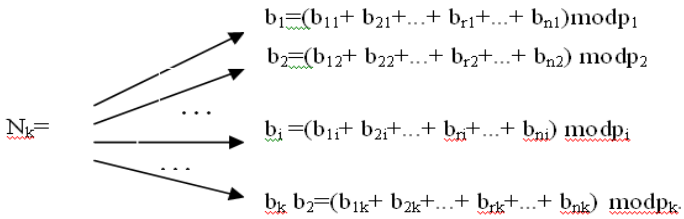


Рис. 9.7. Процес розмежування 32-х розрядного процессора.

Таким чином пряме перетворення РСЗК отримає вигляд :



При цьому, математичні операції над числами в РСЗК можуть бути розмежовані по кожному із фрагментів процесора, що забезпечує ще більш глибокий рівень розпаралелення обробки інформації, а відповідно підвищення швидкодії процесора СЗК.

Таким чином у загальному вигляді зворотне перетворення РСЗК аналітично описується виразом:

$$N_k = res\sum_{r=1}^{n}\sum_{i=1}^{k} res(b_{1i} + b_{2i} + ... + b_{ri} + ... + b_{ni}) \bmod P_i \cdot B_i \bmod P. \qquad (9.35)$$

Застосування міжбазисного перетворення при такому розмежуванні базису Радемахера дозволяє схемотехнічно реалізувати міжбазисне перетворення на основі дешифратора по відповідному модулю.

В задачах цифрового опрацювання сигналів широко використовуються 16-бітні та 24-бітні сигнальні процесори. Ці процесори виконують більше ніж 90% базових операцій додавання, множення та порівняння чисел, при використанні процедур цифрової згортки, фільтрації, цифрової томографії та інше.

Тому можливості реалізації швидкодіючого процесора в РСЗК розглянемо на основі 16-бітного процесора.

Діапазон кодування чисел РСЗК розраховується згідно виразу:

$$P = \prod_{i=1}^{k} p_i \text{ , де } 0 \le N_k \le P-1,$$

N_k – двійковий код числа, p_i – система взаємопростих модулів, а розрядність процесора

$$n_k = \hat{E}[\log_2 (P-1)].$$

З структури розмежованого процесора зрозуміло, що вона потребує обчислення залишків для кожного компонента згідно виразу:

$$b_{ij} = resN_{ij} (\bmod\ p_i).$$

При цьому, процедура обчислення загального залишку виконується згідно виразу:

$$b_i = res(b_{i1} + b_{i2} + ... + b_{in}) \bmod\ p_i.$$

При 4-бітній розрядності компонентів 16-бітного процесора

оптимальний набір модулів з максимальним діапазоном кодування задається масивом чисел:

$$p_i=(3\ ,5\ \ ,7\ ,8\ \ ,11\ ,13),$$

що відповідає діапазону кодування:

$$P=120120.$$

Суттєвою перевагою РСЗК є значне спрощення алгоритму переводу чисел з базису Крестенсона в базис Галуа, що схемотехнічно реалізується за допомогою матричних дешифраторів. В табл. 9.2 приведений приклад реалізації дешифратора 4-бітних компонентів 16-бітного процесора залишкових класів по модулю 5.

Таблиця 9.2.

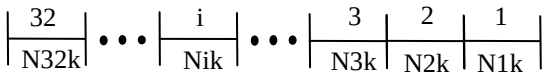
Схема обчислення залишку по модулю 5.

Модуль p_i	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15		Число вентилів
5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2^3	79
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	2^2	
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	2^1	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	2^0	
																	$b_{ij} = resN_{ij} (\mathrm{mod}\ 5)$	
																	2^2	
																	2^1	
																	2^0	

Незважаючи на можливості міжбазисного перетворення Радемахера–Крестонсона на основі дешифратора при степені розмежування двійкових чисел на N – розрядів ($N=2^k$, k=2, 4,…) таке розмежування характеризується недоліком, оскільки при виконанні арифметичних операцій в РСЗК необхідно реалізувати наскрізні переноси між N – розрядними групами бітів базису Радемахера.

9.6.3. Бінарно-розмежована СЗК.

При бінарному розмежуванні двійкових чисел базису Радемахера, тобто k=0, структура розмежування має наступний вигляд:



В результаті такого розмежування двійкового числа (X_{n-1}, X_{n-2}, …,X_i ,… ,X_1 ,X_0) формується матриця залишків кожного *i*-того розряду у системі взаємопростих модулів P_1, P_2, ... P_j, ... P_k (табл.9.3).

Таблиця 9.3.

Матриця залишків числа X.

	X_{n-1}	X_{n-2}	…	X_i	…	X_1	X_0
P_1	$b_{n-1,1}$	$b_{n-2,1}$	…	$b_{i,1}$	…	$b_{1,1}$	$b_{0,1}$
P_2	$b_{n-1,2}$	$b_{n-2,2}$	…	$b_{i,2}$	…	$b_{1,2}$	$b_{0,2}$
P_3	$b_{n-1,3}$	$b_{n-2,3}$	…	$b_{i,3}$	…	$b_{1,3}$	$b_{0,3}$
P_4	$b_{n-1,4}$	$b_{n-2,4}$	…	$b_{i,4}$	…	$b_{1,4}$	$b_{0,4}$
…	…	…	…	…	…	…	…
P_j	$b_{n-1,j}$	$b_{n-2,j}$	…	$b_{i,j}$	…	$b_{1,j}$	$b_{0,j}$
…	…	…	…	…	…	…	…
P_k	$b_{n-1,k}$	$b_{n-2,k}$	…	$b_{i,k}$	…	$b_{1,k}$	$b_{0,k}$

Для переходу в базис Крестенсона над елементами стрічок матриці поданої в табл. 9.3 виконується наступна операція:

$$res(b_{n-1,j} + b_{n-2,j} + ... + b_{i,j} + ... + b_{1,j} + b_{0,j}) \bmod P_i \qquad (9.36).$$

Для підвищення швидкості модульної операції (9.36) доцільно застосувати пірамідальний алгоритм сумування згідно рис.9.8.

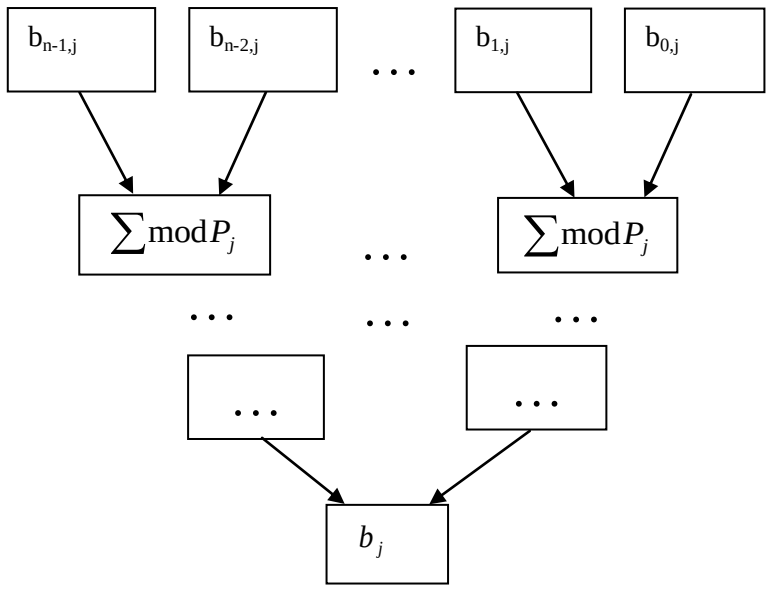


Рис. 9.8. Пірамідальний алгоритм сумування залишків в РСЗК.

Швидкодія такого пірамідально-модульного суматора розраховується за формулою:

$$m = \log_2 n^i,$$

n – розрядність процесора базису Радемахера.

Висока швидкодія такого компонента міжбазисного перетворення Радемахера–Крестенсона потребує великої кількості суматорів в залежності від розрядності процесора число яких розраховується за формулою:

$$S = n + n/2 + n/4 + ... + n/n.$$

Таким чином загальний об'єм апаратного обладнання даного міжбазисного перетворення можна оцінити згідно виразу

$$Q = K \cdot S,$$

де K – число взаємопростих модулів базису Крестенсона.

Об'єм мікроелектронного обладнання, міжбазисного перетворювача (МБП), можна суттєво зменшити на основі паралельної інформаційної технології та структури показаної на рис.9.9.

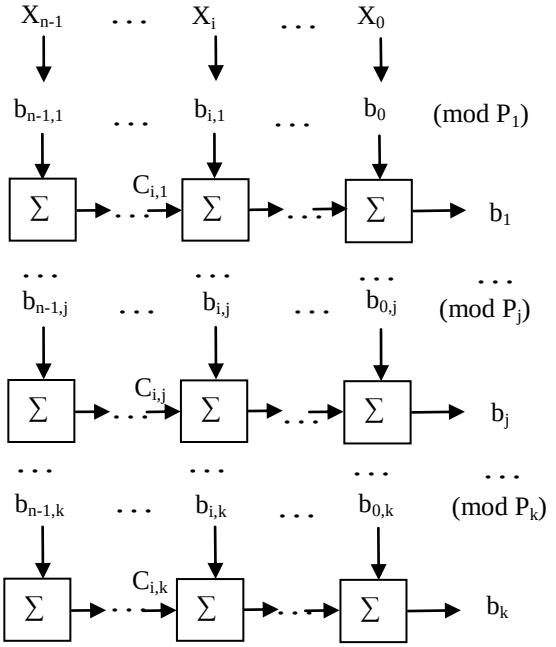


Рис. 9.9. Структура міжбазисного перетворювача Радемахера–Крестенсона.

9.6.4. Оцінка швидкодії сумматора по модулю *Pj*.

Згідно архітектури пірамідального та лінійного міжбазисних перетворень по модулю P_j, показаних архітектур на рис.9.8 та 9.9, порівняння їх характеристик приведено на рис.9.10, звідки видно, що лінійна архітектура потребує в два рази менше обладнання по відношенню до пірамідальної.

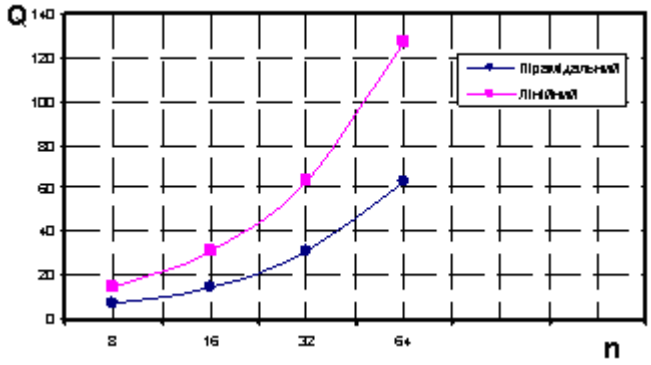


Рис. 9.10. Об'єм мікроелектронного обладнання міжбазисного перетворення Радемахера–Крестенсона.

Результати аналізу швидкодії двох досліджуваних архітектур міжбазисного перетворення при унітарному кодуванні залишків (рис. 9.11) розраховується згідно виразів:

$$V_p = \frac{1}{2 + \log_2 n};\ V_l = \frac{1}{n};$$

де n – число розрядів процесора; V_p – швидкодія пірамідального МБП; V_l – швидкодія лінійного МБП.

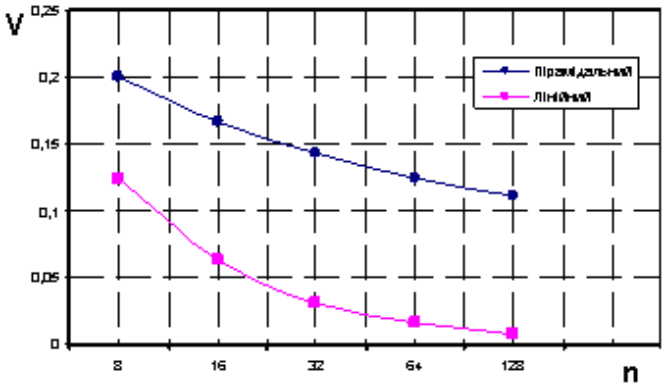


Рис. 9.11. Швидкодія міжбазисного перетворення Радемахера–Крестенсона.

9.6.5. Оцінка складності арифметики у базисах Радемахера, Крестенсона та Галуа.

Порівняльна оцінка функціональних можливостей арифметики названих ТЧБ подана в табл. 9.4.

Таблиця 9.4.

Функціональні можливості ТЧБ.

№	Базові операції	Радемахер	Крестенсон	Галуа	БРСЗК
1	Додавання	*2nv*	*v*	*3v*	*v*
2	Зсув	*v*	-	*2v*	*v*
3	Множення	2*v*(2n+1)	*v*	?	?
4	Рівності	*v*	*v*	*v*	*v*
5	Знакова(старшинства)	*nv*	?	?	*nv*
6	Віднімання	*(3n+5)v*	?	?	2nv
7	Ділення	n^2v	?	-	?
8	Модульна	n^2v	2*nv*	?	2*nv*

В табл.9.4. *n* - розрядність представлення чисел, *v* – часова складність виконання операцій.

Оцінка існування та складності алгоритмів міжбазисних перетворень ТЧБ приведена в табл. 9.5.

Таблиця 9.5.

Міжбазисні перетворення досліджуваних ТЧБ.

Теоретико-числові базиси	Алгоритм міжбазисних перетворень
Радемахера-Крестенсона	$N_k = (a_{n-1},..., a_i,.., a_0); a_i \in \overline{0,1}; N_k = \sum_{i=0}^{n-1} a_i \cdot 2^i;$ $N_k = \begin{matrix} \rightarrow b_1 \\ \rightarrow b_i \\ \rightarrow b_k \end{matrix}$ $b_i = resN_k(\mathrm{mod} p_i);$ $N_k = a_i p_i + b_i,\ P = \prod_{i=1}^{k} p_i;$ $0 \le N_k \le P,\ p_i \ne p_j.$
Радемахера-Галуа	$N_k \to \sum_{i=0}^{n-1} i \to G_0 G_1 ... G_{i-1}; G_0 = (11...1); G_{i+1} = G_i \oplus G_{i-n}; i \in \overline{0, n-1}.$
Крестенсона-Радемахера	$N_k = res \sum_{i=1}^{k} b_i \cdot B_i (\mathrm{mod}\ P),\ B_i = \frac{P}{p_i} \cdot m_i \equiv 1 (\mathrm{mod}\ P_i) \cdot$
Крестесона-Галуа	$N_k = (b_1,.., b_i,.., b_k) \to \sum_{i=0}^{n-1} i \to G_i, ... G_{i-n};$ $G_0 = (11...1); G_{i+1} = G_i \oplus G_{i-n}; i \in \overline{0, n-1}.$

Галуа-Радемахера	$G_{i,}G_{i-1}..G_{i-n} \to \sum_{i=0}^{n-1} i \to$ $N(a_{n-1},...,a_i,..,a_0) \to N_k; a_i \in \overline{0,1}.$
Галуа-Крестенсона	$G_{i,}G_{i-1}...G_{i-n} \to \sum_{i=0}^{n-1} i \to N_k = (b_1, b_2,..., b_i,..., b_k).$

Аналіз сучасного стану реалізації базисних функцій ТЧБ орієнтованих на створення високопродуктивних спецпроцесорів опрацювання великорозрядних чисел дозволяє зробити висновок, що успішний розвиток теорії ТЧБ Крестенсона-Галуа, з метою реалізації всіх базових функцій арифметики процесорів, є доцільним та перспективним.

Необхідно зауважити, що досконала та розмежована форми СЗК є особливо перспективними для створення високопродуктивних мультибазисних процесорів. При цьому новим досягненням в галузі теорії систем числення є розробка двовимірного матричного представлення базису Радемахера.

Сумісне використання двовимірного базису Радемахера, розмежованої, цілочисельної та досконалої нормалізованої форм СЗК дозволяє реалізувати виключно високопродуктивні мультибазисні процесори згідно алгоритму, представленого на рис.9.12.

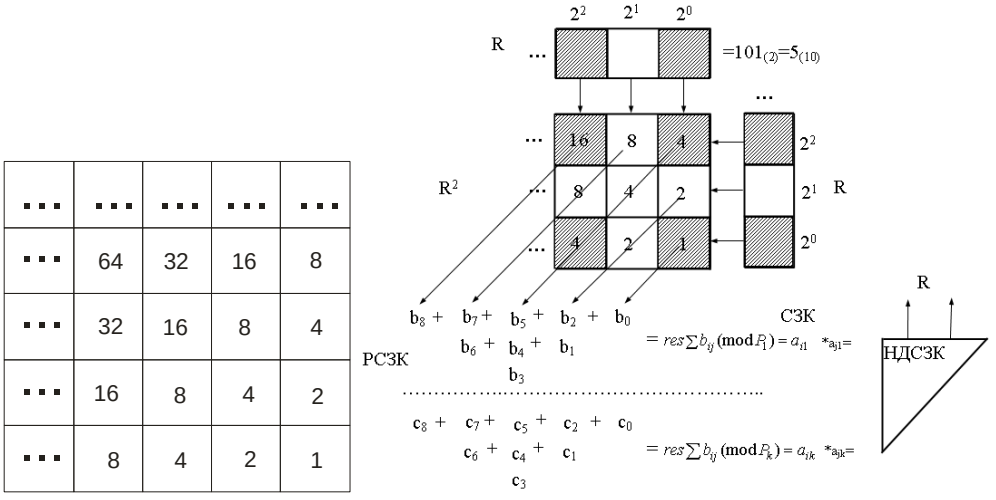


Рис 9.12. Мультибазисні перетворення Радемахера-Крестенсона-Галуа (R-C-G).

9.7. Оптимізація міжбазисного перетворення Галуа-Крестенсона.

Міжбазисне перетворення Радемахера-Галуа та Крестенсона-Галуа може виконуватися на основі проміжного перетворення в Унітарний базис методом одностороннього наближення (рис.9.14).

В той же час такий спосіб міжбазисного перетворення характеризується найбільшою часовою складністю і відповідно низькою швидкодією його реалізації за допомогою спецпроцесорів чи програмних засобів.

Методи адаптивного сходження та інкрементної збіжності за ключем більш ефективні, проте є множина кодових послідовностей, які за допомогою адаптивного сходження декодувати неможливо, а метод збіжності за ключем повністю не використовує інформаційну потужність ключа.

В результаті аналізу структури та особливостей розміщення елементів кодових послідовностей Галуа у фрагментах двійкових рекурентних кодів поля Галуа С.І.Мельничуком запропоновано метод позиційної збіжності за ключем.

Суть методу полягає в проведенні перебору фрагментів кодової послідовності від прийнятої (m_{var}) до ключової (m_{const}) інкрементним або декремнтним зсувом зі змінним кроком (k), величина якого визначається позицією останнього біту, що не співпадає в поточному та ключовому фрагменті і опосередковано від довжини породжуючого ключа (L) коду Галуа:

$$k = f(m_{\text{var}}, m_{const}).$$

Величина зміщення кроку (k) відносно поточного кодового фрагменту визначається на основі побітного порівнювання елементів прийнятого фрагменту послідовності (m_{var}) з елементами ключа (m_{const}). З метою оптимізації порівнювання доцільно здійснювати з кінця кодового фрагменту зменшуючи (k) при рівності відповідних елементів. У випадку нееквівалентності $(m_{var})_k$ та (m_{const}) на основі (k) генерується новий кодон Галуа, після чого порівняння повторюється.

Описана процедура проводиться поки ($k \neq 0$), до співпадання усіх елементів $(m_{var})_k$ та (m_{const}). Напрям зміщення, в кінець чи початок, визначає інкрементний чи декрементний закони сходження до ключа. Реалізація псевдопаралельного сходження, тобто одночасно інкрементного та декрементного зміщення, здійснюється аналогічно до методу псевдопаралельного наближення

Доцільно зазначити, що описаний метод ефективний тільки у випадку використання монотонного породжуючого ключа, тобто коли усі його елементи (біти) однакові “0” чи “1”. Якщо ключем обрано іншу послідовність то декодування здійснюється до моменту досягнення монотонного фрагменту,

після чого необхідно виконати операцію віднімання (чи додавання) величини його зміщення відносно вибраного ключа

Декодування методом інкрементної позиційної збіжності за ключем на прикладі поля Галуа $GF\binom{4}{2}$ зі стартовим кодоном (1111) та довжиною породжуючого ключа $L=4$ біти подано на рис.9.13.

№	Код	k_i
0	1111	0
1	1110	4
2	1101	3
3	1010	4
4	0101	3
5	1011	2
6	0110	4
7	1100	4

№	Код	k_i
8	1001	3
9	0010	4
10	0100	4
11	1000	4
12	0001	3
13	0011	2
14	0111	1
0	1111	0

Рис. 9.13. Міжбазисне перетворення Галуа-Радемахера методом інкрементної збіжності за ключем.

Порівняльні характеристики по кількості необхідних операцій порівняння методів одностороннього наближення, адаптивного сходження, збіжності за ключем та методом позиційної збіжності за ключем подано на рис. 9.14.

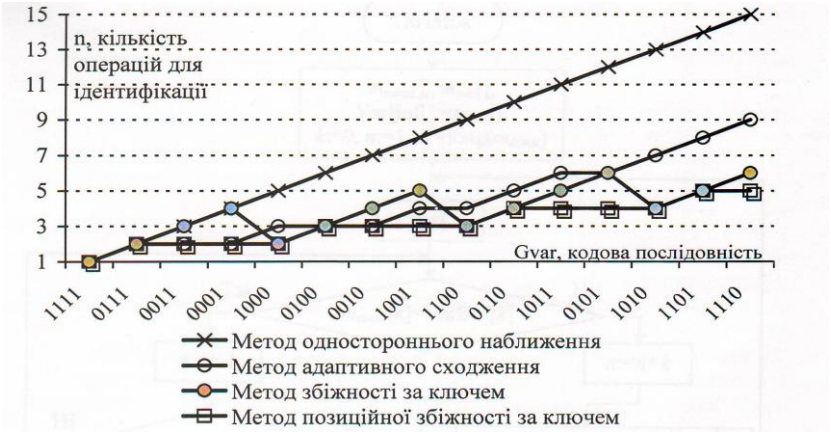


Рис.9.14. Характеристики часової складності між базисного перетворення Галуа-Радемахера.

Як можна побачити повна (побітна) перевірка поточного та базового (ключового) фрагментів здійснюється лише один раз - при досягненні ключа, в усіх інших випадках ця операція припиняється раніше, що фактично не реалізовано в жодному з вище згаданих методів. Такий підхід дозволяє

відмовитись від математичних операцій, а також дещо зменшити кількість операцій порівняння, що виконуються в процесі декодування за рахунок переходів, як показано на рис.9.13.

Декодування рекурентних кодів поля Галуа різних довжин, в порівнянні з методом одностороннього наближення, методом позиційної збіжності за ключем дозволяє зменшити кількість операцій порівняння на 6-67%. Методи збіжності за ключем та адаптивного сходження, в аналогічних умовах, забезпечують зменшення лише на 6-60% та 6-40% відповідно.

Блок-схему алгоритму декодування запропонованого методу подано на рис. 9.15.

В програмній реалізації методу збіжності за ключем, функцію визначення наступної послідовності (фрагменту коду) для порівняння з ключовою *f(k, VarBuf[])* найдоцільніше зробити циклічною - послідовно формувати *k* біт коду без проведення порівнювання з $m_{const}[]$.

Утворення кожного наступного елементу (біту) кодової послідовності можна реалізувати за наступними аналітичними виразами:

$$X_i = X_{i-d} \oplus X_{i-h}, \text{àáî} X_i = X_{i-d} \overline{\oplus} X_{i-h},$$

де X_i - наступний біт коду Галуа; *d, h* - величини зсуву для формуючого ключа.

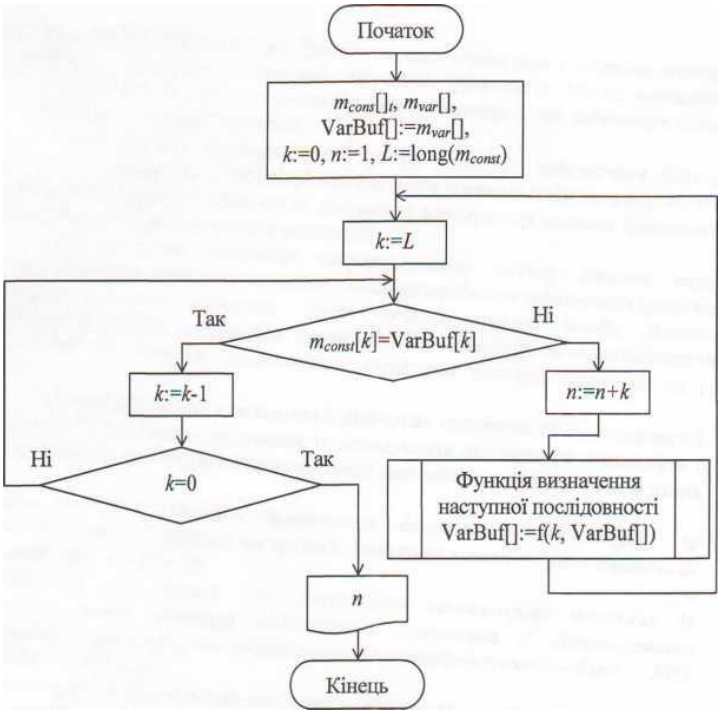


Рис.9.15. Блок-схема алгоритму міжбазисного перетворення Галуа-Крестенсона.

В методах одностороннього та псевдопаралельного наближення кількість операцій порівняння відповідає загальній довжині пошуку. В методах адаптивного сходження та збіжності за ключем кількість таких операцій менша за рахунок використання переходів. Проте операція порівняння в згаданих методах здійснюється над усіма елементами отриманих фрагментів. Метод позиційної збіжності за ключем дозволяє уникнути зайвих перевірок, що забезпечує більш ефективне використання переходів.

Таким чином, описаний метод дозволяє забезпечити більшу ефективність за рахунок оптимізації використання операцій побітного порівняння і, як наслідок, отримати збільшення швидкості декодування кодових фрагментів рекурентних послідовностей Галуа при програмній реалізації.

9.8. Теорія арифметичних операцій в кодах поля Галуа.

У кінцевих полях Галуа правила виконання арифметичних операцій базуються на властивостях ортогональності за симетрії індекса і аргумента у просторі дискретно-постійних функцій ($\varphi(z)$). Таким чином, функції Галуа утворюють базис асимптотично ортогональних рядів, які представляють системи булевих функцій і визначаються правилами додавання, віднімання, множення і ділення над відповідними поліномами по модулю Р.

При цьому операція множення реалізується у вигляді згортки двох поліномів:

$$A(x)=a_{к-i}x^{к-i}+a_{к-2}x^{к-2}+\ldots+a_1x_1+a_0;$$
$$H(x)=h_{z-1}x^{z-2}+h_{z-2}\cdot x^{z-2}+\ldots+h_1x_1+h_0.$$

Операція згортки послідовно виконується починаючи із старших розрядів за к-тактів.

Результат згортки подається у вигляді:

$$A(x)\cdot H(x)= a_{к-i}\, h_{z-1}x^{k+z-2}+(a_{к-2}\cdot h_{z-2}+ a_{к-i}\cdot h_{z-2})\cdot x^{k+z-3}+(a_{к-3}\cdot h_{z-1}+ a_{к-2}\cdot h_{z-2}+ +a_{к-1}\cdot h_{z-3})x^{k+z-4}+\ldots+(a_0\cdot h_2+ a_1h_1+ a_0h_0)x^2+(a_0h_1+ a_1h_0)x+ a_0h_0.$$

Перевагою такої рекурсивної процедури є проста реалізація процесора на регістрах зсуву. Але його швидкодія невелика. Досягнути більшої швидкодії в арифметиці Галуа дозволяє метод безпосереднього обчислення логічного добутку поліномів на основі матричного логічного процесора та відповідних апаратних засобів.

Автором розроблення теоретична основа та запропонований метод реалізації арифметики Галуа на основі каскадного кодування. Суть рішення полягає у тому, що кожен елемент поля в кільці $GF(_n^m)$ представляється у

вигляді логічного виразу рекурсії відповідної операції над попередніми елементами.

Наприклад, у полі Галуа $GF(2^4)$, з ключем $G_{i+1}= G_i \oplus G_{i-4}$, послідовність елементів додавання $a_0, a_2,\ldots a_{15}$ (1111010110010000), які кодують у базисі Галуа відповідні числа 0,1,2,…15 записується у вигляді:

b_1, b_2, b_3, b_4, $b_1 \oplus b_4$, $b_1 \oplus b_2 \oplus b_4$, $b_1 \oplus b_2 \oplus b_3 \oplus b_4$, $b_1 \oplus b_2 \oplus b_3$, $b_2 \oplus b_3 \oplus b_4$, $b_1 \oplus b_3$, $b_2 \oplus b_4$, $b_1 \oplus b_3 \oplus b_4$, $b_1 \oplus b_2$, $b_2 \oplus b_3$, $b_3 \oplus b_4$,Ø.

Особливістю такої паралельної арифметики у ТЧБ Галуа є різна форма подання двох операндів. Перший операнд записується у вигляді коду залишків по відповідному модулю P, а другий операндр задається у вигляді відповідного логічного рівняння над залишками b_1, b_2,… b_4 , причому цей вираз визначає логічні операції по модулю P над кодом першого операнду при обчисленні залишків коду результату сумування кодів Галуа.

Покажемо це на прикладі. Нехай маємо операнди:

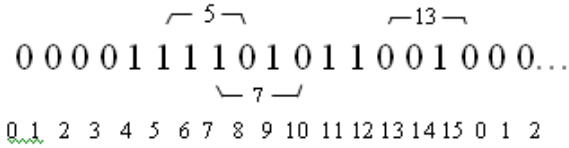
$X = 5_{(10)}, Y = 7_{(10)}$. Обчислимо їх суму, використовуючи КПГ $GF(2^4)$, тобто

	b_1	b_2	b_3	b_4
X=	1	0	1	1
Y=	$b_1 \oplus b_2 \oplus b_3$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2$	$b_2 \oplus b_4$
X+Y=	0	0	0	1

Це відповідає коду Галуа числа $12_{(10)}$.

У загальному випадку нумерацію чисел, які відповідають КПГ можна виконувати будь-яким чином.

Наприклад:



У цьому випадку операція додавання двох чисел $X = 5_{(10)}$, .то $Y = 7_{(10)}$ буде виконуватися наступним чином

	b_1	b_2	b_3	b_4
X=	1	1	0	1
+				
Y=	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3$	$b_2 \oplus b_4$	$b_1 \oplus b_3 \oplus b_4$
X+Y=	0	0	1	0

Очевидно, що приведена арифметика коду Галуа у полі $GF(^4_2)$, однозначно реалізує арифметику 16-ї системи числення з цифрами 0, 1, 2,…A, B, C, D, E, F і може бути ефективно використана при асемблерному програмуванні адресних кодів памяті, на основі 16-ї системи числення.

Важливими перевагами приведеної операції додавання чисел у кодах Галуа по відношенню до виконання операцій додавання двійкових чисел у базисі Радемахера є наступні:

- відсутність наскрізних переносів між розрядами коду Галуа і можливість розпаралелення та паралельного виконання операцій у всіх розрядах одночасно;
- однаковість числа розрядів представлення чисел у базисах Радемахера та Галуа;
- зменшення апаратної складності та підвищення швидкодії процесорів у базисі Галуа порівняно з процесорами у базисі Радемахера та Крестенсона.

Таким чином у загальному вигляді математична операція сумування чисел в кодах Галуа за довільним модулем P виконується наступним чином:

Операнди представляються поліномами $C_{(x)} = \sum_{i=0}^{n-1} c_i x_i \bmod P,$ де

$$C_{(x)} = (a_{n-1}d_{n-1}^{n-1} + a_{n-2}d_{n-2}^{n-2} + \ldots + a_i d_1^{n-1} + a_i d_0^{n-1})x^{n-1} \bmod P + (a_{n-1}d_{n-1}^{n-2} + a_{n-2}d_{n-1}^{n-2}$$
$$+_{n-2}\, d_{n-2}^{n-2} + \ldots + a_1 d_1^{n-2} + a_0 d_0^{n-2})x^{n-2} \bmod P + \ldots + (a_{n-1}d_{n-1}^{1} + a_{n-2}^{2}d_{n-2}^{1} + \ldots$$
$$+ a_1^2 d_1^1 + a_0 d_0^1)x \bmod P + (a_{n-1}d_{n-1}^{0} + a_{n-2}d_{n-2}^{0} + \ldots + a_1 d_1^0 + a_0 d_0)x \bmod P.$$

Для спрощення виразів модульної арифметики при обчислення $C(x)$ доцільно скористатися рівняннями:

$(Pb_i) \bmod P = 0; [(P+1)b_i \bmod P = b_i],$ які слідують з теорії чисел.

Розглянемо приклад операції сумування у полі $GF(^2_3)$, яке описує розряди вісімкової системи числення.

З врахуванням останніх рівнянь, складемо таблицю кодів $GF(^2_3)$, згідно ключа $G_{i+1} = (G_i \oplus G_{i-1}) \bmod 3$ та виразів їх рекурсій

A=8	$GF\begin{pmatrix}2\\3\end{pmatrix}$	$b_i(\mathrm{mod}3)$
0	2	b_1
1	2	b_2
2	X = 1	$b_1 \oplus b_2$
3	0	$b_1 \oplus 2b_2$
4	Y = 1	$2\,b_1$
5	1	$2b_2$
6	2	$2b_1 \oplus 2b_2$
7	0	$2\,b_1$

Нехай $X = 2_{(3)}, Y = 4_{(3)}$, тоді

X=	b_1	b_2
+	1	0
Y=	$2b_1$	$2b_2$
X+Y=	2	0

Це відповідає числу $6_{(8)}$

Операція множення чисел $X \cdot Y$ в кодах поля Галуа може бути виконана двома способами:

1. Шляхом *x*-кратного додавання чисел *Y* при *Y*>*X* або *Y*-кратного додавання чисел *X* при *X*>*Y*;
2. Шляхом послідовного додавання подвоєних значень числа X згідно двійкового представлення числа Y у базисі Радемахера.

Очевидно, що можлива реалізація процесорів у базисі Крестенсона – Галуа в системі взаємо-простих модулів $(p_1 p_2 \cdots p_n)$ при обчисленні кореляційних та спектральних функцій та цифрової фільтрації сигналів, які містять базові операції сумування та множення.

РОЗДІЛ 10

ФУНДАМЕНТАЛЬНІ ПРИКЛАДНІ ЗАДАЧІ У БАЗИСІ КРЕСТЕНСОНА

10.1. Найбільший спільний дільник. Алгоритм Евкліда.

Знаходження найбільшого спільного дільника (НСД) є важливою фундаментальною задачею теорії чисел, успішне вирішення якої дозволяє вдосконалити алгоритми широкого класу прикладних задач, особливо задач захисту інформаційних потоків в комп'ютерних системах з використанням асиметричної криптографії (алгоритмів RSA, Рабіна, Ель-Гамаля, електронного цифрового підпису, дослідження порядку еліптичної кривої за допомогою алгоритму Шуфа). Це зумовлено необхідністю використання, як правило, взаємно простих чисел, НСД яких дорівнює 1.

В зв'язку з цим актуальною проблемою досліджень є розробка теоретичних основ пошуку НСД з використанням теоретико-числових базисів Радемахера та Крестенсона, застосування яких дозволяє зменшити часову складність.

Найбільш розповсюдженим методом знаходження НСД є один з найдавніших математичних алгоритмів – алгоритм Евкліда, згідно якого для знаходження НСД двох чисел необхідно декілька разів від більшого числа відняти менше, поки різниця не стане меншою від'ємника. Тоді цю ж саму процедуру потрібно виконати з від'ємником та різницею. Процес віднімання буде тривати до тих пір, поки від'ємник та різниця не стануть однакові. Оскільки числа, над якими виконуються операції, на кожному кроці зменшуються, то такий процес не може тривати нескінченно, а закінчиться через деяке число кроків.

Сучасний математичний запис алгоритму Евкліда має такий вигляд: для будь-якого $a>b=r_0$, де a і b –цілі числа, виконується система рівнянь

$$\begin{aligned} & a = r_0 \cdot q_1 + r_1,\ 0 \le r_1 < r_0; \\ & r_0 = r_1 \cdot q_2 + r_2,\ 0 \le r_2 < r_1; \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ & r_{n-2} = r_{n-1} \cdot q_n + r_n,\ 0 \le r_n < r_{n-1}; \\ & r_{n-1} = r_n \cdot q_{n+1} + 0. \end{aligned} \qquad (10.1)$$

НСД *(a, b)* дорівнюватиме r_n, тобто останньому ненульовому члену послідовності r_i. Оскільки $r_0>r_1>r_2>\ldots>r_n>0$, то даний процес закінчиться щонайменше через b кроків, тобто потрібно виконати b ділень з остачею. Однак дана оцінка є незадовільною, оскільки для знаходження НСД двох чисел (або перевірки їх на простоту), які використовуються в сучасних

асиметричних криптосистемах, затрачається дуже великий обсяг часу. Для знаходження НСД за допомогою алгоритму Евкліда потрібно виконати не більше, ніж *5k* операцій ділення з остачею, де *k* – кількість цифр в десятковому записі числа a. Кількість кроків не перевищує $2\cdot\log_2 b+1$. Інша оцінка випливає з теореми Ламе, згідно якої кількість кроків алгоритму Евкліда не перевищує $\left[\log_{\Phi}\left(\sqrt{5}a\right)\right]-2$, де $[\alpha]$–верхнє ціле α; $\Phi=\left(1+\sqrt{5}\right)/2$ – більший корінь характеристичного рівняння послідовності Фібоначчі.

Незважаючи на вказані оцінки та простоту програмної реалізації алгоритму Евкліда, його часова складність залишається великою, оскільки операція ділення є досить трудомісткою. Розрахунки показують, що пошук НСД з використанням алгоритму Евкліда в базисі Радемахера характеризується часовою складністю щонайменше $17{,}5\cdot n(n+1)^2$, де *n* – розрядність числа a в двійковому коді.

Іншим недоліком алгоритму Евкліда є послідовне виконання операції ділення одна за одною, тобто неможливість розпаралелення його роботи.

Перспективним напрямком вдосконалення досліджуваних алгоритмів є розробка теорії їх реалізації на основі розмежованої системи числення залишкових класів.

Для зменшення часової складності потрібно розробляти нові високопродуктивні алгоритми знаходження НСД за допомогою розмежованої системи числення та базису Крестенсона.

10.1.1. Алгоритм Евкліда в розмежованій системі числення.

Нехай потрібно знайти НСД чисел *a* і *b*:

$$a=a_{n-1}2^{n-1}+\ldots+a_i2^i+\ldots+a_12+a_0 \qquad (10.2)$$
$$b=b_{n-1}2^{n-1}+\ldots+b_i2^i+\ldots+b_12+b_0, \qquad (10.3)$$

при чому $a>b=r_0$; $a,b=0,1$.

Виходячи з (10.1), $r_1=a\bmod b=(a_{n-1}2^{n-1}+\ldots+a_i2^i+\ldots+a_12+a_0)\bmod b=\left(\sum_{i=0}^{n-1}\left(a_i\,2^i \bmod b\right)\right)\bmod b=\left(\sum_{i=0}^{n-1}\left(a_i r_{1i}\right)\right)\bmod b$, де $r_{1i}=2^i \bmod b$. Це означає, що шуканий залишок дорівнюватиме сумі тих степенів двійки, для яких відповідно $a_i=1$. Слід зазначити також, що два послідовні значення r_{1i} та $r_{1\,i+1}$ пов'язані рекурентним співвідношенням $r_{1\,i+1}=(2\cdot r_{1i})\bmod b$. Для знаходження залишку за модулем *b* не обов'язково виконувати ділення з остачею, а можна обмежитися відніманням: якщо $r_{1\,i+1}<b$, то воно залишається незмінним, в іншому випадку $r_{1\,i+1}=r_{1\,i+1}-b$. Найпростіше

реалізувати описаний крок алгоритму Евкліда в розмежованій системі числення за допомогою таблиці 10.1.

Таблиця 10.1.

Таблиця знаходження залишку a mod b.

a_{n-1}	a_{n-2}	…	a_i	…	a_2	a_1	a_0
r_{1n-1}	r_{1n-2}	…	$r_{1\,i}$	…	r_{12}	r_{11}	r_{10}

Згідно таблиці 10.1, r_1 шукається як сума r_{1i} за модулем b, над якими у верхньому рядку розміщено 1, тобто $r_1 = \left(\sum_{i=0}^{n-1} r_{1i}\right) \bmod b$ при умові $a_i=1$.

Аналогічно будуємо таблицю 10.2.

Таблиця 10.2.

Таблиця знаходження залишку b mod r_1.

$b_{n-1}=r_{0\,n-1}$	$b_{n-2}=r_{0\,n-2}$	…	$b_i=r_{0\,i}$	…	$b_2=r_{0\,2}$	$b_1==r_{0\,1}$	$b_0=r_{0\,0}$
r_{2n-1}	r_{2n-2}	…	$r_{2\,i}$	…	r_{22}	r_{21}	r_{20}

Відповідно $r_2 = \left(\sum_{i=0}^{n-1} r_{2i}\right) \bmod r_1$ при умові $b_i=1$.

Узагальнюючи отримані результати, запишемо вираз для знаходження будь-якого залишку:

$$r_j = \left(\sum_{i=1}^{n-1} r_{j-2i} r_{ji}\right) \bmod r_{j-1},$$

де $r_{j-2\,i}=0,1$; $r_{ji}=2^i \bmod r_{i-1}$.

Відмітимо, що кількість кроків стандартного алгоритму Евкліда та алгоритму Евкліда в розмежованій системі числення однакові. Однак часова складність виконання кожного кроку істотно зменшується і становить $\log_2 n/2$. Загальна часова складність алгоритму Евкліда в розмежованій системі числення оцінюється виразом $O(17{,}5n(\log_2 n/2))$. Крім того, він виключає можливість розпаралелення.

Запропонований алгоритм знаходження НСД ґрунтується на пошуку залишків від ділення чисел a і b $(a > b)$ в розмежованій системі числення на всі прості числа до $\sqrt{b}$.

Спільним дільником чисел a та b буде модуль p_j^k, який шукається з умови:

$$\left(\sum_{i=0}^{n-1} a_{ij}\right) \bmod p_j^k = \left(\sum_{i=0}^{n-1} b_{ij}\right) \bmod p_j^k = 0, \qquad (10.4)$$

де $a_{ij} = a_i \cdot 2^i \bmod p_j^k$, $b_{ij} = b_i \cdot 2^i \bmod p_j^k$, p_j – просте число, менше $\sqrt{b}$, k =1,2,3, ... – степінь p_j.

Слід зазначити, що при $k=1$ і виконанні (10.4) перевіряється та ж умова при k=2 і т.д. Таким чином враховується спільний дільник, який є степенем простого числа. Шуканий найбільший спільний дільник знаходиться, як добуток отриманих за допомогою (10.4) спільних дільників.

Запропонований алгоритм характеризується логарифмічною часовою складністю $O(m \cdot \log_2 n)$, де $m = \int_2^{\sqrt{b}} \frac{dt}{\ln t}$ – кількість простих чисел в діапазоні від 2 до $\sqrt{b}$. Крім того, він дозволяє розпаралелити виконання всіх операцій по кожному модулю та виконати факторизацію чисел *a* та *b*.

Запропонований вище алгоритм можна суттєво удосконалити шляхом скорочення кількості модулів (тобто кількості кроків), за якими потрібно шукати залишки. Нехай маємо систему модулів, яка складається з простих чисел $p_1, p_2, p_3, ...$, менших $\sqrt{b}$, і для деякого p_j^k виконується умова (10.4). Наступний крок полягає у послідовній перевірці умови (10.4) для модуля $\left(p_j^k p_{j+1}^{k_1}\right)$, $k_1=1,2,3,...$; $i=1,2,....$

Таким чином, при послідовному помноженні модулів отримуємо, що НСД$(a, b) = \prod_{j=1}^{s} p_j^k$, для яких виконується умова (10.4).

В порівнянні з попереднім, даний алгоритм використовує меншу кількість кроків, однак він не піддається розпаралеленню і не вирішує задачу факторизації чисел. Загальна часова складність удосконаленого алгоритму, яка визначається сумою складностей основних операцій, становить $O\left(\log_2 n \cdot \left(\log_2 n + \frac{n}{2} + k \cdot \log_2 n\right) + \log_2 \frac{n}{2}\right)$.

Приклади застосування алгоритмів пошуку НСД.

Нехай потрібно обчислити НСД(3843, 1449).

1. Стандартний алгоритм Евкліда:

3843=1449·1+945;

1449=945·1+504;

945=504·1+441;

504=441·1+63;

441=63·7+0;

Отже, НСД(3843, 1449)=63.

Алгоритм Евкліда в розмежованій системі числення зручно представити у вигляді таблиці 10.3.

Таблиця 10.3.

Алгоритм Евкліда в розмежованій системі числення.

1	3843	1	1	1	1	0	0	0	0	0	0	1	1
2	$2^i \bmod 1449$	599	1024	512	256	128	64	32	16	8	4	2	1
3	1449		1	0	1	1	0	1	0	1	0	0	1
4	$2^i \bmod 945$		79	512	256	128	64	32	16	8	4	2	1
5	945			1	1	1	0	1	1	0	0	0	1
6	$2^i \bmod 504$			8	256	128	64	32	16	8	4	2	1
7	504				1	1	1	1	1	1	0	0	0
8	$2^i \bmod 441$				256	128	64	32	16	8	4	2	1
9	441				1	1	0	1	1	1	0	0	1
10	$2^i \bmod 63$				4	2	1	32	16	8	4	2	1

З рядка 2 видно, що $(599+1024+512+256+2+1) \bmod 1449=945$.

З рядка 4: $(79+256+128+32+8+1) \bmod 945=504$.

З рядка 6: $(8+256+128+32+16+1) \bmod 504=441$.

З рядка 8: $(256+128+64+32+16+8) \bmod 441=63$.

З рядка 10: $(4+2+32+16+8+1) \bmod 63=0$.

Таким чином можна отримати НСД, уникнувши громіздкої операції ділення.

Пошук НСД в базисі Крестенсона.

Дану задачу також зручно представити у вигляді таблиці 10.4, врахувавши, що $\sqrt{1449} \approx 38$.

Таблиця 10.4.

Знаходження залишків по простих модулях.

1		2048	1024	512	256	128	64	32	16	8	4	2	1
2	3843	1	1	1	1	0	0	0	0	0	0	1	1
3	1449		1	0	1	1	0	1	0	1	0	0	1
4	$2^i \bmod 2$	0	0	0	0	0	0	0	0	0	0	0	1
5	$2^i \bmod 3$	2	1	2	1	2	1	2	1	2	1	2	1
6	$2^i \bmod 5$	3	4	2	1	3	4	2	1	3	4	2	1
7	$2^i \bmod 7$	4	2	1	4	2	1	4	2	1	4	2	1
8	$2^i \bmod 11$	2	1	6	3	7	9	10	5	8	4	2	1
9	$2^i \bmod 13$	7	10	5	9	11	12	6	3	8	4	2	1
10	$2^i \bmod 17$	8	4	2	1	9	13	15	16	8	4	2	1
11	$2^i \bmod 19$	15	17	18	9	14	7	13	16	8	4	2	1
12	$2^i \bmod 23$	1	12	6	3	13	18	9	16	8	4	2	1
13	$2^i \bmod 29$	18	9	19	24	12	6	3	16	8	4	2	1

продовження таблиці 10.4

14	$2^i \bmod 31$	2	1	16	8	4	2	1	16	8	4	2	1
15	$2^i \bmod 37$	13	25	31	34	17	27	32	16	8	4	2	1
16	$2^i \bmod 9$	5	7	8	4	2	1	5	7	8	4	2	1
17	$2^i \bmod 27$	23	25	26	13	20	10	5	16	8	4	2	1

З таблиці 10.4 шукаються залишки по простих модулях.

Рядок4: $3843 \bmod 2=1;\ 1449 \bmod 2=1$;

Рядок 5: $3843 \bmod 3=(2+1+2+1+2+1) \bmod 3=0;\ 1449 \bmod 3=(1+1+2+2+2+1) \bmod 3=0$;

Рядок 6: $3843 \bmod 5=(3+4+2+1+2+1) \bmod 5=3;\ 1449 \bmod 3=(4+1+3+2+3+1) \bmod 5=4$;

Рядок 7: $3843 \bmod 7=(4+2+1+4+2+1) \bmod 7=0;\ 1449 \bmod 7=(2+4+2+4+1+1) \bmod 7=0$;

Рядок 8: $3843 \bmod 11=(2+1+6+3+2+1) \bmod 11=4;\ 1449 \bmod 11=(1+3+7+10+8+1) \bmod 3=8$;

Рядок 9: $3843 \bmod 13=(7+10+5+9+2+1) \bmod 13=8;\ 1449 \bmod 3=(10+9+11+6+8+1) \bmod 13=6$;

Рядок 10: $3843 \bmod 17=(8+4+2+1+2+1) \bmod 17=1;\ 1449 \bmod 17=(4+1+9+15+8+1) \bmod 17=4$;

Рядок 11:$3843 \bmod 19=(15+17+18+9+2+1) \bmod 19=5;\ 1449 \bmod 19=(17+9+14+13+8+1) \bmod 19=5$;

Рядок 12: $3843 \bmod 23=(1+12+6+3+2+1) \bmod 23=2;\ 1449 \bmod 23=(12+3+13+9+8+1) \bmod 23=0$;

Рядок 13: $3843 \bmod 29=(18+9+19+24+2+1) \bmod 29=15;\ 1449 \bmod 29=(9+24+12+3+8+1) \bmod 29=28$;

Рядок 14: $3843 \bmod 31=(2+1+16+8+2+1) \bmod 31=30;\ 1449 \bmod 31=(1+8+4+1+8+1) \bmod 31=23$;

Рядок 15: $3843 \bmod 37=(13+25+31+34+2+1) \bmod 37=32;\ 1449 \bmod 37=(25+34+17+32+8+1) \bmod 37=6$.

Дані розрахунки показують, що спільними простими дільниками є числа 3 та 7. Для знаходження НСД потрібно перевірити їх степені:

Рядок 16: $3843 \bmod 3^2=(5+7+8+4+2+1) \bmod 3^2=0;\ 1449 \bmod 3^2=(7+4+2+5+8+1) \bmod 3^2=0$;

Рядок 17:$3843 \bmod 3^3=(23+25+26+13+2+1) \bmod 3^3=9;\ 1449 \bmod 3^3=(25+13+20+5+8+1) \bmod 3^3=18$.

Число $7^2=49>38$ і його можна не перевіряти. Отже, НСД$(3843, 1449)=3^2 \cdot 7=63$. Даний метод дозволяє провести факторизацію чисел, наприклад $1449=3^2 \cdot 7 \cdot 23$. Крім того, обчислення можна виконувати паралельно по різних модулях.

Удосконалений алгоритм пошуку НСД в базисі Крестенсона.

Будуємо таблицю 10.5.

Таблиця 10.5.

Знаходження залишків в удосконаленому алгоритмі.

1		2048	1024	512	256	128	64	32	16	8	4	2	1
2	3843	1	1	1	1	0	0	0	0	0	0	1	1
3	1449		1	0	1	1	0	1	0	1	0	0	1
4	$2^i \bmod 2$	0	0	0	0	0	0	0	0	0	0	0	1
5	$2^i \bmod 3$	2	1	2	1	2	1	2	1	2	1	2	1
6	$2^i \bmod 9$	5	7	8	4	2	1	5	7	8	4	2	1
7	$2^i \bmod 27$	23	25	26	13	20	10	5	16	8	4	2	1
8	$2^i \bmod 45$	23	34	17	31	38	19	32	16	8	4	2	1
9	$2^i \bmod 63$	32	16	8	4	2	1	32	16	8	4	2	1
10	$2^i \bmod 441$	284	142	71	256	128	64	32	16	8	4	2	1
11	$2^i \bmod 693$	662	331	512	256	128	64	32	16	8	4	2	1

Аналізуємо таблицю 10.5.

Рядок 4: $3843 \bmod 2=1;\ 1449 \bmod 2=1$;

Рядок 5: $3843 \bmod 3=(2+1+2+1+2+1) \bmod 3=0;\ 1449 \bmod 3=(1+1+2+2+2+1) \bmod 3=0$;

Рядок 6: $3843 \bmod 9=(5+7+8+4+2+1) \bmod 9=0;\ 1449 \bmod 9=(7+4+2+5+8+1) \bmod 9=0$;

Рядок 7: $3843 \bmod 27=(23+25+26+13+2+1) \bmod 27=9;\ 1449 \bmod 27=(25+13+20+5+8+1) \bmod 27=18$;

Рядок 8: $3843 \bmod 45=(23+34+17+31+2+1) \bmod 45=18;\ 1449 \bmod 45=(34+31+38+32+8+1) \bmod 3=8$;

Рядок 9: $3843 \bmod 63=(32+16+8+4+2+1) \bmod 63=0;\ 1449 \bmod 63=(16+4+2+32+8+1) \bmod 63=0$;

Рядок 10: $3843 \bmod 441=(284+142+71+256+2+1) \bmod 441=315;\ 1449 \bmod 441=(142+256+128+32+8+1) \bmod 441=126$;

Рядок 11: $3843 \bmod 693=(662+331+512+256+2+1) \bmod 693=378;\ 1449 \bmod 693=(331+256+128+32+8+1) \bmod 693=63$.

З розрахунків також випливає, що НСД$(3843, 1449)=63$.

Крім того, зазначимо, що в двох останніх алгоритмах не обов'язково шукати залишки від обох чисел a та b. Досить знайти залишки від меншого числа і тільки при їх рівності 0 перевіряти друге число.

Складність запропонованих алгоритмів визначається часовою складністю наступних операцій:

1) знаходженні залишків a_j, b_j чисел X, Y по простих модулях $p_j^{m_j}$ для яких виконується умова $a_j = b_j = 0$.

2) обчислення добутку модулів $Z = НСД(X,Y) = \prod_{j=1}^{k} p_j^{m_j}$.

В таблиці 10.6 подано оцінки часової складності основних операцій алгоритму пошуку НСД в базисі Крестенсона, що дозволяє зробити порівняльний аналіз з вищенаведеними алгоритмами пошуку найбільшого спільного дільника.

Таблиця 10.6.

Часова складність основних операцій алгоритму пошуку НСД в базисі Крестенсона та його удосконалення.

№	Основні операції	Часова складність
1.	$p_j^{m_j}$	$O(n) = \left(log_2 n \cdot \left(log_2 n + \frac{n}{2} \right) \right)$
2.	$a_j^{(m)} = res\left(\sum_{i=1}^{n-1} a_{ij} (\mathrm{mod} p_j^{m}) \right)$ $b_j^{(m)} = res\left(\sum_{i=1}^{n-1} b_{ij} (\mathrm{mod} p_j^{m}) \right)$	$O(n) = \left(log_2 n/2 \right)$
3.	$Z = \prod_{j=1}^{k} p_j^{m_j}$	$O(n) = \left(k \cdot log_2 n \right)$

де k - кількість модулів для яких виконується умова $a_j = b_j = 0$.

З врахуванням даних табл. 10.6, загальна часова складність запропонованого алгоритму пошуку НСД в базисі Крестенсона, та його удосконалення буде визначатися сумою складностей основних операцій, а саме: $O(n) = \left(log_2 n \cdot \left(log_2 n + \frac{n}{2} + k \cdot log_2 n \right) + n \cdot log_2 n/2 \right)$ і

$OЗ(n) = \left(log_2 n \left(log_2 n + k \cdot log_2 n + \frac{n}{2} \right) + \frac{n}{2} \cdot log_2 \frac{n}{2} \right)$ відповідно. На рис.10.1 показані графіки, які характеризують складності існуючого та запропонованих алгоритмів в залежності від розрядності компонентів Z .

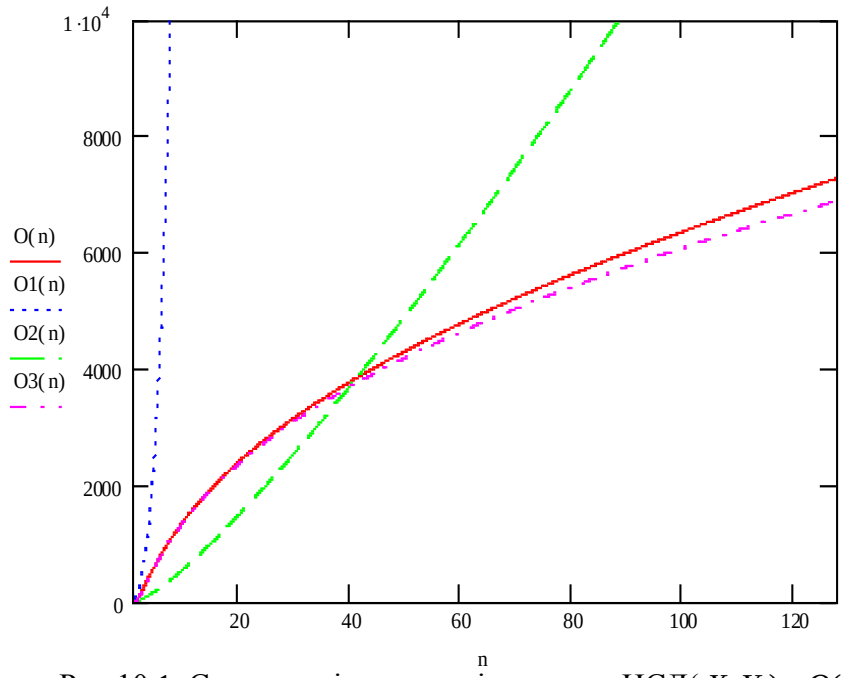


Рис.10.1. Складності алгоритмів пошуку НСД(X, Y) , $O(n)$ - складність алгоритму пошуку НСД в базисі Крестенсона, $O1(n)$ – алгоритму Евкліда, $O2(n)$ – удосконалення реалізації алгоритму Евкліда з використанням розмежованої системи числення Радемахера – Крестенсона, $O3(n)$ – удосконалений алгоритм пошуку НСД в базисі Крестенсона.

Результати досліджень показали, що для пошуку НСД двох чисел існуючий алгоритм Евкліда, який традиційно використовується для пошуку НСД для чисел великої розрядності при сучасному рівні комп'ютерної техніки стає практично нездійсненний. Чисельний експеримент оцінки складностей запропонованих алгоритмів пошуку НСД показує, що в діапазоні двійкових розрядів від 0 до 40 бітів, слід використовувати удосконалення реалізації алгоритму Евкліда з використанням розмежованої системи числення Радемахера – Крестенсона, а при збільшенні розрядності чисел потрібно застосовувати алгоритм пошуку НСД в базисі Крестенсона та його удосконалення.

10.2. Китайська теорема про залишки (КТЗ).

Перетворення Китайської теореми про залишки (КТЗ) є фундаментальною основою вирішення широкого класу задач теорії чисел, а також прикладних задач інженерії та інформатики.

Незважаючи на свою простоту та древню історію, КТЗ продовжує представляти себе у новому світлі і відкривати нові перспективи свого застосування, особливо у математиці, інформатиці (машинна арифметика), криптографії тощо. Побудова непозиційної системи числення в часових системах (системи залишкових класів) для виконання операцій з великими числами, дискретне перетворення Фур'є, генерування таємних ключів в асиметричних криптосистемах, зв'язок з класичною поліноміальною інтерполяційною теорією, багатовимірні обчислення, можливість зведення вивчення кільця лишків за модулем m (де m – довільне ціле число) до вивчення кільця лишків за модулем p^s (p – просте число), дослідження алгебраїчних кілець, можливість арифметичної самокорекції кодів та розпаралелення обчислень, визначення послідовності великого числа зразків ДНК – ось далеко не повний перелік сучасного застосування КТЗ.

КТЗ є одним з найдавнішим, але важливим часовим алгоритмом. Ще в першому столітті нашої ери китайський математик Сунь-Цзи придумав загадку, якою було покладено початок модулярній арифметиці: знайти число, яке при діленні на 3 дасть в остачі 2, на 5 – 3, на 7 – 2. Крім того, він показав у частковому випадку еквівалентність розв'язку системи модулярних рівнянь і розв'язку одного модулярного рівняння.

Протягом майже двох тисяч років КТЗ постійно вдосконалювалася та розвивалася. Зокрема, в XIII столітті інший китайський математик Цань Цзю-Шао розв'язав наведену вище задачу. У XVIII столітті німецький математик Л.Ейлер навів загальне формулювання та доведення КТЗ, а К.-Ф. Гаус істотно розвинув його в своїх знаменитих „Арифметичних дослідженнях".

І, нарешті, в середині XX століття чеські учені М.Валах та А.Свобода запропонували використати древню китайську ідею на новому технічному рівні, створивши перші модулярні електронно-часові машини „Епос" та „Епос–2". Їх ідеї підтримали радянські та українські вчені Ф.Лукін, І.Акушський, Д.Юдіцький, Є.Адріанов, В.Амербаєв, Я. Николайчук та інші.

Слід зазначити, що на даний час існує декілька еквівалентних формулювань КТЗ. Найбільш поширене з них таке: якщо натуральні числа $p_1, p_2, \ldots, p_k$ попарно взаємно прості, то для будь-яких цілих $r_1, r_2, \ldots, r_k$, таких що $0 \le r_i < p_i$ існує число N, яке при діленні на p_i дає залишок r_i при всіх $i=1, 2, ..., k$; більше того, якщо існує два таких числа N_1 та N_2, то $N_1 \bmod P = N_2 \bmod P$, де $P = \prod_{i=1}^{k} p_i$.

Дану теорему можна представити у вигляді системи порівнянь:

$$\begin{cases} N \bmod p_1 = r_1 \\ N \bmod p_2 = r_2 \\ \ldots\ldots\ldots\ldots\ldots \\ N \bmod p_i = r_i \\ \ldots\ldots\ldots\ldots\ldots \\ N \bmod p_k = r_k. \end{cases} \qquad (10.5)$$

Шукане число обчислюється за формулою:

$$N = \left(\sum_{i=1}^{k} M_i m_i r_i \right) \bmod P, \qquad (10.6)$$

де $M_i = \dfrac{P}{p_i} = p_1 p_2 \ldots p_{i-1} p_{i+1} \ldots p_k$, $m_i = M_i^{-1} \bmod p_i$.

Відмітимо, що на даний час відомі три способи пошуку оберненого елемента m_i:

1) перевірка шляхом послідовної підстановки чисел натурального ряду у формулу, поки не буде виконуватись умова $M_i m_i \bmod p_i = 1$;

2) використовуючи функцію Ейлера, можна знайти $m_i = M_i^{-1} \bmod p_i = M_i^{\varphi(p_i - 1)} \bmod p_i$;

3) за допомогою розширеного алгоритму Евкліда.

Однак кожен з цих способів характеризується значною часовою складністю при виконанні ділень з остачею, піднесення до степеня, знаходженні функції Ейлера (факторизації p_i). Причому всі ці операції повинні виконуватися над дуже великими числами, що приводить до переповнення розрядної сітки сучасних потужних часових засобів.

Автором запропонована досконала форма системи залишкових класів, у якій підбір модулів такий, що $M_i \bmod p_i = 1, m_i = 1$. В подальшому було розвинуто дану теорію та розроблено її модифікований варіант, коли $M_i \bmod p_i = \pm 1, m_i = \pm 1$, тобто відповідний підбір модулів дозволяє уникнути процедури знаходження оберненого елемента. Недолік даного методу полягає в тому, що не завжди є можливість вибору відповідної системи модулів.

10.2.1. Теоретичні основи алгоритмів перетворення КТЗ в базисі Радемахера-Крестенсона.

Для спрощення розглянемо два взаємно прості модулі $p_1 < p_2$. Нехай потрібно знайти число N, яке при діленні на p_1 дає залишок r_1, а при діленні на p_2 – залишок r_2, що еквівалентно такій системі порівнянь:

$$\begin{cases} N \bmod p_1 = r_1 \\ N \bmod p_2 = r_2. \end{cases} \tag{10.7}$$

Розв'язок (9.7) можна представити у такому вигляді:

$$N = \left(r_1 p_2 \left(p_2^{-1} \bmod p_1\right) + r_2 p_1 \left(p_1^{-1} \bmod p_2\right)\right) \bmod p_1 p_2 . \tag{9.8}$$

Для знаходження $p_2^{-1} \bmod p_1$ представимо p_2 у двійковій формі: $p_2=a_{n-1}2^{n-1}+a_i \cdot 2^i+\ldots+a_1 \cdot 2^1+ +a_0 \cdot 2^0$, де a_i=0,1, і сформуємо таблицю 10.7.

Щоб знайти елемент p_{2i} необхідно попередній елемент p_{2i-1} домножити на 2 (дописати в кінці 0 у двійковому записі) і порівняти з модулем p_1. остаточна формула для p_{2i} матиме такий вигляд:

Таблиця 10.7.

Знаходження залишків степенів двійки.

2^{n-1}	2^{n-2}	…	2^i	…	2	1
a_{n-1}	a_{n-2}	…	a_i	…	a_1	a_0
$p_{2\,n-1}$	$p_{2\,n-2}$	…	$p_{2\,i}$	…	$p_{2\,1}$	$p_{2\,0}$

$$p_{2i} = \begin{cases} 2 \cdot p_{2\,i-1}, \; 2 \cdot p_{2\,i-1} < p_1 \\ 2 \cdot p_{2\,i-1} - p_1, \; 2 \cdot p_{2\,i-1} \geq p_1. \end{cases} \tag{10.9}$$

Отже, уникнувши громіздкої операції ділення, знаходимо залишок $p_3=p_2 \bmod p_1$. Він буде дорівнювати сумі тих p_{2i}, для яких відповідні a_i=1. Тоді $p_2^{-1} \bmod p_1 = p_3^{-1} \bmod p_1$.

Для знаходження оберненого елемента знову ж шукаємо залишок $p_1 \bmod p_3=p_{10}$. Оскільки $p_{10} \neq 0$, то далі виконується така послідовність кроків: $(p_1+1) \bmod p_3=(p_{10}+1) mod \; p_3=p_{11}$; $(2p_1+1) \bmod p_3=(p_{11}+p_1) \bmod p_3=p_{12}$; …. ; $(i \cdot p_1+1) \bmod p_3=(p_{1i-1}+p_1) \bmod p_3=p_{1i}$; … . Описану послідовність продовжуємо до тих пір, поки p_{1i} не стане рівним нулю. Зазначимо, що процедура знаходження p_{1i} аналогічна визначенню залишку p_3.

Обернений елемент $p_2^{-1} \bmod p_1$ дорівнюватиме результату ділення $(i \cdot p_1+1)$ на p_3. Для уникнення цієї громіздкої операції потрібно описаним вище методом знайти залишки $b_i=(i \cdot p_1+1) \bmod p_3 \cdot q_i^s$, де q_i пробігає послідовність простих чисел, $p_3 \cdot q_i^s < (i \cdot p_1+1)$, s=1, 2, … , причому s збільшується на 1, коли b_i=0. Шукане обернене число p_2^{-1} буде дорівнювати добутку тих q_i^s, для яких відповідні b_i=0.

За аналогічним алгоритмом шукається $p_1^{-1} \bmod p_2$.

Наступним кроком є обчислення добутків трьох множників за модулем P у кожному доданку (10.9). Операцію множення пропонуємо виконати матричним методом, що істотно зменшує часову складність.

Розглянемо два числа $x=x_{n-1}2^{n-1}+\ldots+x_i2^i+\ldots+x_12+x_0$ та $y=y_{n-1}2^{n-1}+\ldots+y_j2^j+\ldots+y_12+y_0$, де $x_i, y_j=0, 1$, n-розрядність модуля P. Для знаходження результату їх множення за модулем P побудуємо матрицю, представлену в таблиці 9.8, де $c_{ij}=2^{i+j} \bmod P$.

Таблиця 10.8.

Матриця для множення двох n-розрядних двійкових чисел.

	b_{n-1}	…	b_j	…	b_1	b_0
a_{n-1}	$c_{n-1\ n-1}$	…	$c_{n-1\ j}$	…	$c_{n-1\ 1}$	$c_{n-1\ 0}$
…	…	…	…	…	…	…
a_i	$c_{i\ n-1}$	…	c_{ij}	…	c_{i1}	c_{i0}
…	…	…	…	…	…	…
a_1	$c_{1\ n-1}$	…	c_{1j}	…	c_{11}	c_{10}
a_0	$c_{0\ n-1}$	…	c_{0j}	…	c_{01}	c_{00}

Добуток чисел x та$\cdot y$ отримуємо за формулою:

$$x \cdot y = \left(\sum_{m,k=1}^{n-1} c_{mk} \right) \bmod P, \qquad (10.10)$$

де x_m, $y_k=1$, тобто c_{mk} знаходиться на перетині стовпця та рядка, для яких відповідні x_i та y_j дорівнюють 1.

Останнім кроком знаходження шуканого числа N є визначення суми двох чисел за модулем P.

10.2.2. Застосування запропонованих алгоритмів.

Розглянемо приклад. Нехай потрібно знайти число N, яке при діленні на $p_1=43$ дає остачу $r_1=10$, а при діленні на $p_2=209$ – остачу $r_2=100$ ($P=209\cdot43=8987$).

Знайдемо $p_3=209 \bmod 43$ матричним методом, представленим у таблиці 10.9.

Таблиця 10.9.

Знаходження залишку за модулем.

2^i	128	64	32	16	8	4	2	1
209	1	1	0	1	0	0	0	1
$2^i mod\ 43$	42	21	32	16	8	4	2	1

Отже, $p_3=(42+21+16+1) \bmod 43=37$.

Далі знаходимо $p_3^{-1} \bmod 43 = 37^{-1} \bmod 43$. Для цього шукаємо $43 \bmod 37=6$, додаємо 1 і послідовно додаємо 6, поки в результаті додавання за mod37 не буде 0. Представимо це у вигляді таблиці 10.10.

Таблиця 10.10.

Пошук оберненого елемента за модулем.

i	0	1	2	3	4	5	6
p_{1i}	6	7	13	19	25	31	0

Звідси число $K_1=6\cdot43+1$, яке націло ділиться на 37. Добуток $6\cdot43$ знайдемо також матричним методом, представленим у таблиці 10.11, записавши обидва множники у двійковій формі: $6=(110)_2$; $43=(101011)_2$.

Таблиця 10.11.

Множення двох n-розрядних двійкових чисел

	0	0	0	1	1	0
1	1024	512	256	128	64	32
0	512	256	128	64	32	16
1	256	128	64	32	16	8
0	128	64	32	16	8	4
1	64	32	16	8	4	2
1	32	16	8	4	2	1

Отже, $K_1=6\cdot43+1=(128+64+32+16+8+4+4+2)+1=259$. Діленням можна знайти, що $p_3^{-1} \bmod p_1 = 37^{-1} \bmod 43=259:37=7$. Матричним методом це представлено в таблиці 10.12 (не перевіряючи парного простого числа 2).

Таблиця 10.12.

Пошук оберненого елемента $37^{-1} \bmod 43$.

2^i	256	128	64	32	16	8	4	2	1	
259	1	0	0	0	0	0	0	1	1	
$2^i \bmod 3\cdot37$	34	17	64	32	16	8	4	2	1	*(34+2+1)mod111=37*
$2^i \bmod 5\cdot37$	71	128	64	32	16	8	4	2	1	*(71+2+1)mod185=74*
$2^i \bmod 7\cdot37$	256	128	64	32	16	8	4	2	1	*(256+2+1)mod259=0*

Звідси видно, що $p_3^{-1} \bmod p_1 = 37^{-1} \bmod 43 = 7$. Далі шукається $p_1^{-1} \bmod p_2 = 43^{-1} \bmod 209$. Вище було знайдено, що $209 \bmod 43 = 37$. Будується таблиця 10.13.

Таблиця 10.13.

Пошук оберненого елемента $43^{-1} \bmod 209$ в розмежованому базисі.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
p_{2i}	37	38	32	26	20	14	8	2	39	33	27	21	15	9	3	40	34	28	22
i	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
p_{2i}	16	10	4	41	35	29	23	17	11	5	42	36	30	24	18	12	6	0	

Тоді $K_2 = 36 \cdot 209 + 1 = 7525$ і $p_1^{-1} \bmod p_2 = 43^{-1} \bmod 209 = 7525 : 43 = 175$. Матричним методом виходить аналогічний результат, представлений у таблиці 10.14.

Таблиця 10.14.

Пошук оберненого елемента $43^{-1} \bmod 209$ матричним методом.

2^i	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
7525	1	1	1	0	1	0	1	1	0	0	1	0	1	
$2^i \bmod 3 \cdot 43$	97	113	121	125	127	128	64	32	16	8	4	2	1	$(97+113+121+ +127+64+32+ +4+1) \bmod 129 = 45$
$2^i \bmod 5 \cdot 43$	11	113	164	82	41	128	64	32	16	8	4	2	1	$(11+113+164 +41+64+32+ +4+1) \bmod 215 = 0$
$2^i \bmod 5^2 \cdot 43$	871	973	1024	512	256	128	64	32	16	8	4	2	1	$(871+973+1024+ +256+64+32+4+ +1) \bmod 1075 = 0$
$2^i \bmod 5^3 \cdot 43$	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	$(4096+2048+1024+ +256+64+32+4+ +1) \bmod 5375 = 2150$
$2^i \bmod 5^2 \cdot 7 \cdot 43$	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	(4096+2048+1024+ +256+64+32+4+ +1)mod7525=0

Отже, $p_1^{-1} \bmod p_2 = 43^{-1} \bmod 209 = 5^2 \cdot 7 = 175$. Звідси видно, що шукане число: $N = (209 \cdot 7 \cdot 10 + 43 \cdot 175 \cdot 100) \bmod 8987 = 3235$.

10.2.3. Оцінка та порівняльний аналіз часових складностей відомих та запропонованих алгоритмів.

При перетвореннях згідно КТЗ використовуються такі основні модульні операції:

1) знаходження оберненого елемента;

2) знаходження залишків;

3) операції множення та додавання.

Тому при визначенні часових складностей відомого та запропонованого алгоритмів, які дозволяють виконувати перетворення КТЗ, потрібно враховувати складності вищезазначених операцій, наведені у таблиці 10.15.

Таблиця 10.15.

Часові складності основних операцій КТЗ.

№	Основні операції	Часова складність операцій у запропонованому алгоритмі	Часова складність операцій у класичному алгоритмі
1.	Пошук оберненого елемента	$O(n) = \left(\frac{n^2 \cdot k}{2} \cdot \right)$	$O1(n) = (17,5k \cdot ((n+1)^2 + n^2 + n)$
2.	Пошук залишків	$O(n) = \left(\log_2 n \middle/ 2 \right)$	$O1(n) = ((n+1)^2 + n)$
3.	$\sum_{i=1}^{k} a_i N_i M_i \bmod \cdot n$	$O(n) = \left(\log_2 k \cdot (2 \cdot \log_2{}^2 n + n \right)$	$O1(n) = (k \cdot (2n^2 + n))$

де k - кількість взаємно простих модулів $n = n_1 \cdot n_2 \cdot \ldots \cdot n_k$.

На рисунку 10.2 показано графіки залежності часових складностей від розрядності чисел *n*. З рисунка видно, що використання запропонованого алгоритму, який грунтується на використанні теоретико-числового базису Крестенсона, дозволяє істотно зменшити часову складність КТЗ відносно класичного.

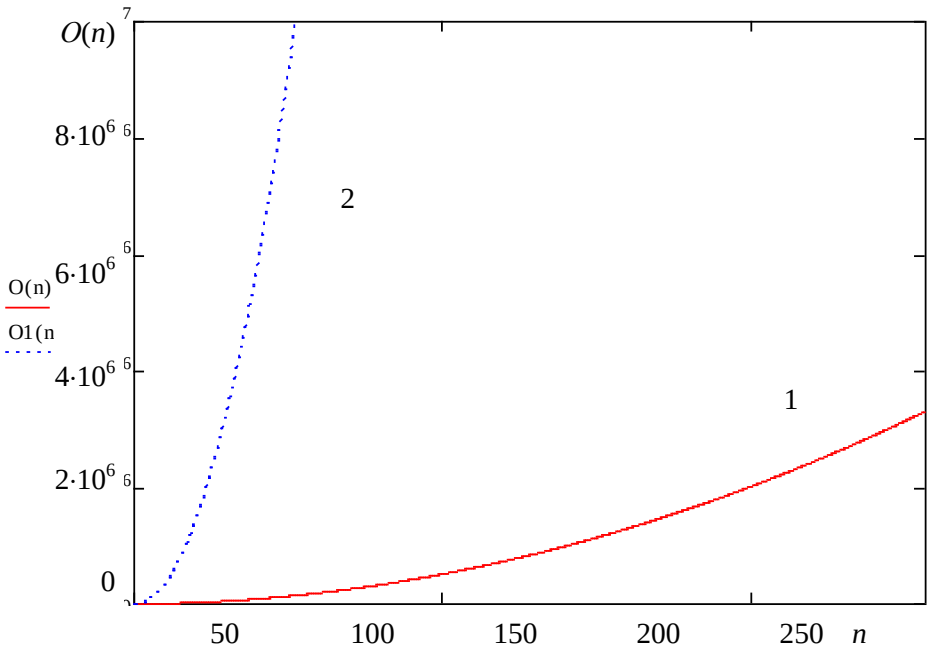


Рис.10.2. Графіки залежності часових складностей від розрядності чисел *n* методом з використанням ТЧБ Радемахера – Крестенсона (1) та класичним (2).

Враховуючи табличні дані, часова складність Китайської теореми про залишки з використанням ТЧБ Радемахера-Крестенсона становить $O(n) = \left(\left(\log_2 k \cdot (2 \cdot \log_2{}^2 n + n) \right) + \frac{n^2 \cdot k}{2} + \left(\log_2 n/2 \right) \right)$, а з використанням класичного алгоритму – $O1(n) = (37k \cdot n^2 + 53{,}5k \cdot n + 17{,}5k + n^2 + 3n + 1))$.

10.2.4. Високопродуктивні алгоритми множення великорозрядних чисел у базисі Крестенсона-Радемахера.

Однією з найважливіших операцій в усіх асиметричних алгоритмах шифрування інформаційних потоків є модулярне множення багато розрядних чисел *a* та *b* розміру *n*. Для підвищення ефективності обчислення $a \cdot b (\mathrm{mod}\, p)$ багатьма авторами запропоновано різні підходи, алгоритми, але в основному з використанням десяткової системи числення з часовою складністю $O(n)=n^2$, що значно збільшує час виконання програми.

Одним з підходів щодо підвищення швидкодії знаходження результату модулярного множення є метод Карацуби, але він практично не використовують через складність його реалізації. В алгоритмі Шенхаге – Штрассена з використанням швидкого перетворення Фур'є і потребує $O(n)=(n\log(n)\log(\log(n))$ бітових операцій.

Тому розробка ефективного алгоритму з використанням теоретико-числового базису Крестенсона для зменшення часової складності формування, збільшення швидкодії на основі матричних моделей та розробка спеціалізованих програмно-апаратних засобів є актуальною задачею.

Операції модулярного множення в базисі Радемахера лежать в основі алгоритмів електронного цифрового підпису, криптографічних протоколів, розв'язування задач часової, прикладної та дискретної математики. Визначення стійкості еліптичних кривих методом пошуку їх порядку за допомогою алгоритму Шуфа тощо. Існуючі алгоритми (стандартний, швидкого множення, Blakey, Монтгомері, бінарні і т.д.) характеризуються значною часовою складністю. Запропонований алгоритм модулярного множення в базисі Крестенсона за допомогою матричних обчислень, дозволить зменшити часову складність.

Відомо, що в базисі Крестенсона будь-яке ціле десяткове число N представляється у вигляді набору найменших невід'ємних залишків від його ділення на фіксовані цілі додатні взаємно прості модулі p_i, причому $0 \le N \le \prod_{i=1}^{n} p_i - 1$. Зворотне перетворення в десяткову систему числення є набагато складнішим. Оскільки операції в комп'ютерних системах виконуються в базисі Радемахера, то постає питання в прямому переході з базису Крестенсона в базис Радемахера і навпаки, що ефективно виконується за допомогою принципової розмежованої форми системи числення в базисах Радемахера-Крестенсона.

Розглянемо два n-розрядних числа $a=a_{n-1}2^{n-1}+\ldots+a_i2^i+\ldots+a_12+a_0$ та $b=b_{n-1}2^{n-1}+\ldots+b_i2^j+\ldots+b_12+b_0$, де a_i, $b_j=0$, 1, n-розрядність модуля p. Для знаходження результату їх множення за модулем p побудуємо матрицю, представлену в таблиці 10.16, де $c_{ij}=2^{i+j} \bmod p$.

Добуток чисел a та·b отримуємо за формулою:

$$a \cdot b = \left(\sum_{m,k=1}^{n-1} c_{mk} \right) \bmod p \ , \qquad (9.10)$$

де a_m, $b_k=1$, тобто c_{mk} знаходиться на перетині стовбця та рядка, для яких відповідні a_i та b_j дорівнюють 1.

Таблиця 10.16.

Матриця множення в базисі Радемахера–Крестенсона.

	b_{n-1}	…	b_j	…	b_1	b_0
a_{n-1}	$c_{n-1\ n-1}$	…	$c_{n-1\ j}$	…	$c_{n-1\ 1}$	$c_{n-1\ 0}$
…	…	…	…	…	…	…
a_i	$c_{i\ n-1}$	…	c_{ij}	…	c_{i1}	c_{i0}
…	…	…	…	…	…	…
a_1	$c_{1\ n-1}$	…	c_{1j}	…	c_{11}	c_{10}
a_0	$c_{0\ n-1}$	…	c_{0j}	…	c_{01}	c_{00}

Модулярне множення здійснюється згідно алгоритму:

Таким чином, отриманий новий алгоритм заміни операції множення, яка має квадратичну часову складність $O1(n) = n^2$, операцією додавання з логарифмічною складністю:

$$O2(n) = \begin{cases} (\log_2 n)^2, & якщо\ \ n < 256 \\ n \cdot (\log_2 n)^2, & в\ \ інших\ \ випадках \end{cases}.$$

Результати дослідження ефективності запропонованого алгоритму приведена на рисунку 10.3.

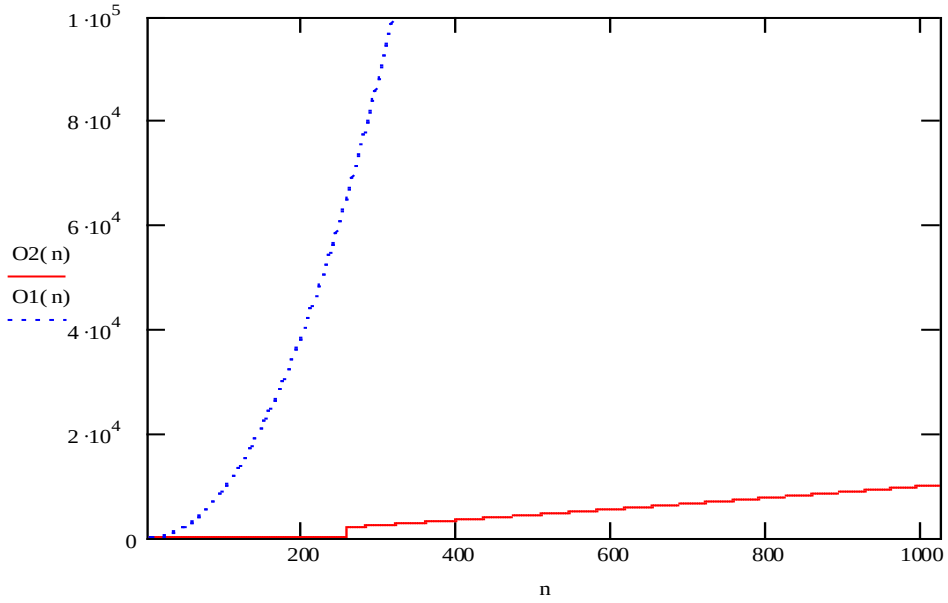


Рис.10.3. Складність операції модулярного множення.

З рисунка 10.3 видно, що при зростанні розрядності компонентів дискретного логарифма n>256 відомий алгоритм не може бути фізично реалізований на даний час з використанням сучасної мікропроцесорної

техніки, особливо з врахуванням наступної операції піднесення до степеня, яка включає операцію множення. А запропонований алгоритм з використанням базису Крестенсона дозволяє реалізувати можливості суттєвого підвищення захисту інформаційних потоків від несанкціонованого доступу, які характеризуються ознаками незворотності, як показано в роботах Николайчука Я.М., повинно бути практично нездійсненна, що строго математично не доведено.

Ефективність запропонованого алгоритму модулярного множення буде $E3(n) = \begin{cases} \frac{n^2}{(\log_2 n)^2}, якщо n < 256 \\ \frac{n^2}{n \cdot \log_2 n}, в інших випадках. \end{cases}$ (рис.9.4)., як співвідношення часових складностей.

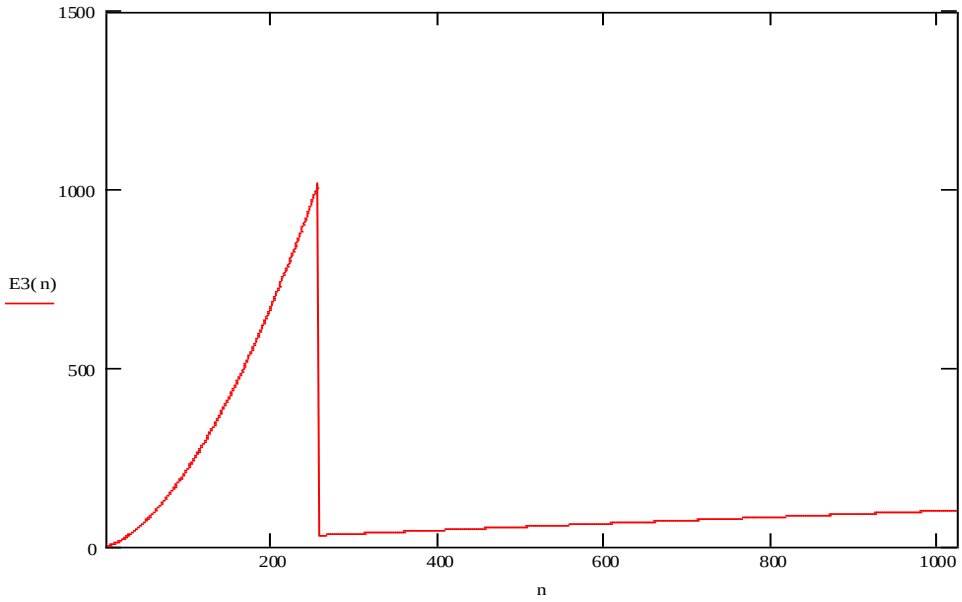


Рис.10.4 Ефективність алгоритму модулярного множення розмірності n.

Результати чисельного експеременту показують, що при розмірності чисел від 0 до 256 бітів ефективність стрімко зростає, за рахунок переходу від двовимірного базису Радемахера, складність операції модулярного множення в якому мають квадратичну залежність, до розмежованої системи числення Радемахера-Крестенсона з логарифмічною складністю. А в діапазоні від 256 до 1024 біти в розмежованій системі числення складність операції множення буде лінійно-логарифмічною, тому що при реалізації

запропонованого алгоритму операція додавання чисел великої розрядності більше 256 біт, яка замінює операцію множення в відомих алгоритмах, здійснюється не за один такт, як при менших розрядах, а за декілька тактів.

Отже, з використанням базису Крестенсона суттєво збільшуються перспективи щодо розробки систем з високим рівнем захисту інформаційних потоків та дозволяє пришвидшити час виконання алгоритму модулярного множення.

10.3. Алгоритм піднесення до високих показників степенів у розмежованій системі числення базису Крестенсона-Радемахера.

Для модулярного експоненціювання $a^x \bmod p$ (вважаємо, що $x \le \varphi(p)$, $\varphi(p)$ – значення функції Ейлера від модуля p) використаємо проміжну матрицю, представлену в табл. 10.17. Її розмірність дорівнює розрядності n модуля p. В стовбцях матриці записано величини $a^{2^i} \bmod p$ в базисі Радемахера, тобто $a_{ij}=0, 1$. Тоді будь-який степінь x можна записати за степенями 2 і шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 10.16. Основними перевагами такого методу є здійснення операцій в системі залишкових класів, а не оперувати з великими числами, що дозволяє пришвидшити алгоритм модулярного експоненціювання.

Таблиця 10.17.

Матриця піднесення до степеня в базисі Радемахера–Крестенсона.

$a_{n-1\ n-1}$	…	$a_{i\ n-1}$		$a_{1\ n-1}$	$a_{0\ n-1}$
…	…	…	…	…	…
$a_{n-1\ j}$	…	$a_{i\ j}$	…	$a_{1\ j}$	$a_{0\ j}$
…	…	…	…	…	…
$a_{n-1\ 1}$	…	$a_{i\ 1}$	…	$a_{1\ 1}$	$a_{0\ 1}$
$a_{n-1\ 0}$	…	$a_{i\ 0}$	…	$a_{1\ 0}$	$a_{0\ 0}$
$a^{2^{n-1}}$	…	a^{2^i}	…	a^{2^1}	a^{2^0}

Отже, для модулярного експоненіювання $a^x \pmod p$ потрібно скористатися алгоритмом піднесення до степеня двійкового числа будь-якої розрядності за модулем p.

Пропонований алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем p дозволяє зменшити часову складність за рахунок заміни операції множення операцією додавання, підвищити швидкодію на 30-40% (рис.10.5), для чисел розрядності менше 256 біт, а в діапазоні від 256 біт на 10%. Чисельний експеримент показав, що

запропонований алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем p в базисі Радемахера–Крестенсона дозволяє зменшити складність з $O3(n) = \frac{n^3}{2}$ до

$$O4(n) = \begin{cases} \log_2 n \cdot \left(\frac{n}{2} \cdot \log_2 n + 1 \right), якщо n < 256 \\ n \cdot \left(\frac{n}{2} \cdot \log_2 n + 1 \right), в інших випадках. \end{cases}$$

в

$$E2(n) = \begin{cases} \frac{n}{(\log_2 n)^2 + 2 \cdot \log_2 n}, для\ n < 256 \\ \frac{n}{(\log_2 n) + 2 \cdot \log_2 n}, 256 \le n \le 1024 \end{cases}$$

разів, що показано на рис. 10.6.

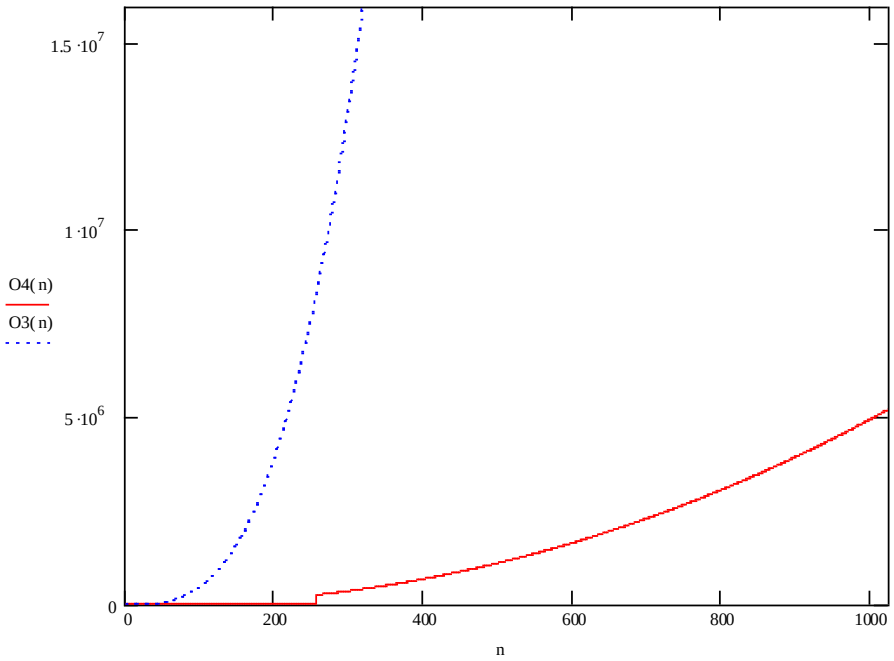


Рис.10.5. Часова складність операції модулярного піднесення до степеня.

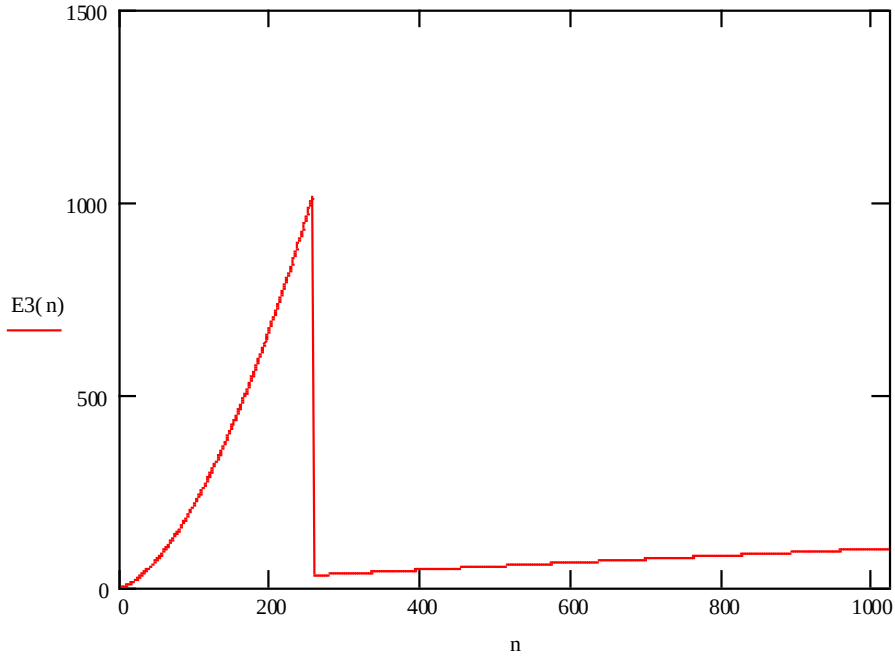


Рис.10.6. Ефективність запропонованого алгоритму.

Дослідження показали, що алгоритм запропонований алгоритм характеризується високою швидкодією та ефективністю для знаходження значення операції піднесення до степеня двійкового числа будь-якої розрядності за модулем *p*. Слід зазначити, що при збільшенні розрядності чисел зменшується ефективність (рис 10.6), бо частина ресурсу комп'ютера буде задіяна на розв'язок службової інформації, що значно збільшує складність і кількість операцій та зменшує швидкодію. Оскільки, операція модулярного експоненіювання є базовою в найбільш поширених системах захисту інформаційних потоків з відкритими ключами (RSA, Ель–Гамаля тощо), визначення стійкості еліптичних кривих методом пошуку їх порядку за допомогою алгоритму Шуфа, то доцільно використовувати розроблений метод в задачах захисту інформаційних потоків на практиці для вдосконалення систем захисту інформаційних потоків.

10.4. Теорія алгоритмів RSA та Ель-Гамаля в розмежованій системі числення Радемахера-Крестенсона.

Аналіз наукових тенденцій розвитку теорії та перспективних інформаційних технологій покращення ефективності опрацювання інформаційних потоків в комп'ютерних мережах, проведений на основі

новітніх публікацій, потребує поглибленого дослідження теоретичних засад базисів Крестенсона та Радемахера. Слід зауважити, що найбільш фундаментально досліджено цілочисельну форму в системі залишкових класів, яка утворюється на основі прямого перетворення ТЧБ Крестенсона.

Тому є доцільним дослідити інші форми систем залишкових класів, які можуть бути використані для реалізації високопродуктивних алгоритмів опрацювання і захисту інформаційних потоків, а також виконати порівняльний аналіз різних ТЧБ з базисом Радемахера, який породжує двійкову систему числення на основі відповідних критеріїв, та дослідити часову складність алгоритмів шифрування інформаційних потоків з використання алгоритмів RSA, Ель-Гамаля в розмежованій системі числення Радемахера-Крестенсона.

Незважаючи на ефективні алгоритми формування структуризованих даних, які реалізуються на основі програмно-апаратних мультибазисних процесорів і забезпечують захист від помилок та певний рівень захисту інформації (ЗІ) від несанкціонованого доступу, який не відповідає умовам сучасного рівня ЗІ. Тому потрібно додаткове опрацювання інформаційних потоків, формування захищених даних, що потребує аналізу і ефективності існуючих методів захисту структуризованих даних на основі відомих алгоритмів RSA, Ель-Гамаля.

10.4.1 Оцінка часових складностей алгоритмів опрацювання інформації в задачах криптографії.

Сучасний розвиток комп'ютерної техніки потребує високого рівня захисту інформації, додаткового опрацювання інформаційних потоків та формування захищених даних. Тому зроблений аналіз ефективності існуючих методів захисту структуризованих даних на основі відомих алгоритмів RSA, Ель-Гамаля, а особливо алгоритмів з використанням математичного апарату еліптичних кривих, говорить, що їх використання є перспективним для вдосконалення захисту інформаційних потоків.

Система захисту інформаційних потоків від несанкціонованого доступу з відкритим ключем RSA базується на задачі множення і розкладу чисел на прості множники, які є часово однонаправленими задачами. В системі RSA кожний з учасників процесу шифрування має в розпорядженні відкритий і закритий (секретний) ключі, кожний з яких складається з пари цілих чисел, які генеруються наступним чином:

1. Генеруються два випадкових простих числа p і q довільного розміру (чим більші, тим краще для стійкості системи захисту інформаційних потоків).

2. Обчислюється $n = p \cdot q$, тобто модуль криптоперетворень з використанням матричних перетворень: подаємо число p і q у вигляді: $p = p_{r-1}2^{r-1} + p_{r-2}2^{r-2} + \ldots + p_1 2^1 + p_0 2^0$,
$q = q_{r-1}2^{r-1} + q_{r-2}2^{r-2} + \ldots + q_1 2^1 + q_0 2^0$, де r- розрядність чисел p і q. Для знаходження результату їх множення побудуємо матрицю, представлену в табл. 10.18, де $m_{ij} = 2^{i+j}$.

Добуток чисел p і q отримуємо за формулою:

$$n = p \cdot q = \sum_{s,k=1}^{r-1} m_{sk} \qquad (10.11)$$

де $p_s, q_k = 1$, тобто m_{sk} знаходиться на перетині стовпця та рядка, для яких відповідні p_i і q_j дорівнюють 1. Це значно зменшить складність алгоритму пошуку модуля криптоперетворень $O(n) = \left(3{,}5n^2 + n - \frac{1}{2}\right)$.

Таблиця 10.18.

Матриця знаходження модуля перетворення в базисі Радемахера–Крестенсона.

	q_{r-1}	…	q_j	…	q_1	q_0
p_{r-1}	$m_{r-1\ r-1}$	…	$m_{r-1\ j}$	…	$m_{r-1\ 1}$	$m_{r-1\ 0}$
…	…	…	…	…	…	…
p_i	$m_{i\ r-1}$	…	m_{ij}	…	m_{i1}	m_{i0}
…	…	…	…	…	…	…
p_1	$m_{1\ r-1}$	…	m_{1j}	…	m_{11}	m_{10}
p_0	$m_{0\ r-1}$	…	m_{0j}	…	m_{01}	m_{00}

3. Аналогічним чином знаходимо значення функції Ейлера від числа n, а саме $\varphi(n) = (p-1)(q-1)$.

4. Вибираємо ціле число $e, 1 < e < \varphi(n)$ - взаємнопросте з $\varphi(n)$. Його доцільно вибирати з найменшою кількістю одиничних бітів в двійковій формі, рекомендуються вибирати числа Ферма 17, 257, 65537… Число e називають відкритою експонентою; час шифрування залежить від швидкодії операції піднесення числа в степінь по модулю, тому малі значення e можуть зменшити стійкість системи захисту інформаційних потоків.

5. Знаходимо секретну експоненту d обернену до числа e за модулем $\varphi(n)$, тобто $d \cdot e \equiv 1 (\mathrm{mod}\varphi(n))$. В класичній літературі знаходження оберненого елемента в залишкових класах обчислюється згідно

розширеного алгоритму Евкліда. Для зменшення складності доцільно скористатися матричним методом:

Розглянемо модуль $\varphi(n)$. Нехай x пробігає зведену систему найменших додатних лишків за модулем $\varphi(n)$: $r_1 = 1, r_2 = 2, r_3 = 3, ..., r_i = i, ..., r_{\varphi(n)-1} = \varphi(n) - 1$.

Тоді для числа $e < \varphi(n)$ добуток $e \cdot x$ теж пробігатиме зведену систему лишків за цим модулем:

$$\begin{cases} e \cdot r_1 \bmod \varphi(n) = c_1 \\ e \cdot r_2 \bmod \varphi(n) = c_2 \\ e \cdot r_3 \bmod \varphi(n) = c_3 \\ \ldots\ldots\ldots\ldots\ldots \\ e \cdot r_i \bmod \varphi(n) = c_i \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ e \cdot r_{\varphi(n)-1} \bmod \varphi(n) = c_{\varphi(n)-1}. \end{cases} \quad . \tag{10.12}$$

Аналогічно для числа d:

$$\begin{cases} d \cdot r_1 \bmod \varphi(n) = g_1 \\ d \cdot r_2 \bmod \varphi(n) = g_2 \\ d \cdot r_3 \bmod \varphi(n) = g_3 \\ \ldots\ldots\ldots\ldots\ldots \\ d \cdot r_i \bmod \varphi(n) = g_i \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ d \cdot r_{\varphi(n)-1} \bmod \varphi(n) = g_{\varphi(n)-1}. \end{cases} \quad . \tag{10.13}$$

З першої системи виберемо перше рівняння, а з другого – рівняння, в якому $c_1 = r_i$: $e \cdot r_1 \bmod \varphi(n) = c_1$

$$d \cdot r_i \bmod \varphi(n) = g_i . \tag{10.14}$$

Перемножимо ці рівняння:

$$er_1 \cdot dr_i \bmod \varphi(n) = c_1 \cdot g_i . \tag{10.15}$$

Враховуючи, що $r_1 = 1, c_1 = r_i$, отримаємо добуток за відповідним модулем:

$$e \cdot d \bmod \varphi(n) = g_i . \tag{10.16}$$

Даним методом можна перемножувати будь-яку кількість множників, підносити до будь-якого степеня, шукати обернені елементи та квадратні

корені. Для цього зручно побудувати таблицю Келі (табл. 10.19) наприклад, для $\varphi(n)$=20, для $p=3, q=11.$

6. В результаті отримуємо пари $P=(e,n)$ - відкритий ключ, $S=(d,n)$ - таємний ключ.

Після генерування ключів, шифрування інформаційних потоків здійснюється наступним чином:

- Беремо $P=(e,n)$ і інформаційний потік у вигляді цілого числа $M \in D \in Z_n, 0<M<n-1.$

Таблиця 10.19.

Таблиця Келі.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	**4**	6	8	10	12	14	16	18	20	22	1	3	5	7	9	11	13	15
3	6	**9**	12	15	18	21	1	4	7	10	13	16	19	22	2	5	8	11
4	8	12	**16**	20	1	5	9	13	17	21	2	6	10	14	18	22	3	7
5	10	15	20	**2**	7	12	17	22	4	9	14	19	1	6	11	16	21	3
6	12	18	1	7	**13**	19	2	8	14	20	3	9	15	21	4	10	16	22
7	14	21	5	12	19	**3**	10	17	1	8	15	22	6	13	20	4	11	18
8	16	1	9	17	2	10	**18**	3	11	19	4	12	20	5	13	21	6	14
9	18	4	13	22	8	17	3	**12**	21	7	16	2	11	20	6	15	1	10
10	20	7	17	4	14	1	11	21	**8**	18	5	15	2	12	22	9	19	6
11	22	10	21	9	20	8	19	7	18	**6**	17	5	16	4	15	3	14	2
12	1	13	2	14	3	15	4	16	5	17	**6**	18	7	19	8	20	9	21
13	3	16	6	19	9	22	12	2	15	5	18	**8**	21	11	1	14	4	17
14	5	19	10	1	15	6	20	11	2	16	7	21	**12**	3	17	8	22	13
15	7	22	14	6	21	13	5	20	12	4	19	11	3	**18**	10	2	17	9
16	9	2	18	11	4	20	13	6	22	15	8	1	17	10	**3**	19	12	5
17	11	5	22	16	10	4	21	15	9	3	20	14	8	2	19	**13**	7	1
18	13	8	3	21	16	11	6	1	19	14	9	4	22	17	12	7	**2**	20
19	15	11	7	3	22	18	14	10	6	2	21	17	13	9	5	1	20	**16**

- Шифрується інформаційний потік, і передається по каналу зв'язку згідно співвідношення $P(M)=M^e \bmod n$, використовуючи алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем *p*. Її розмірність дорівнює розрядності n модуля p. В стовбцях матриці записано величини $M^{2^i} (\mathrm{mod} n)$ в базисі Радемахера, тобто M_{ij}=0, 1

(табл.10.20). Тоді будь-який степінь e можна записати за степенями 2 і шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 10.20.

Таблиця 10.20.

Матриця шифротексту алгоритму шифрування RSA в базисі Радемахера–Крестенсона.

$M_{n-1\ n-1}$	…	$M_{i\ n-1}$		$M_{1\ n-1}$	$M_{0\ n-1}$
…	…	…	…	…	…
$M_{n-1\ j}$	…	$M_{i\ j}$	…	$M_{1\ j}$	$M_{0\ j}$
…	…	…	…	…	…
$M_{n-1\ 1}$	…	$M_{i\ 1}$	…	$M_{1\ 1}$	$M_{0\ 1}$
$M_{n-1\ 0}$	…	$M_{i\ 0}$	…	$M_{1\ 0}$	$M_{0\ 0}$
$M^{2^{n-1}}$	…	M^{2^i}	…	M^{2^1}	M^{2^0}

Для дешифрування використовується таємний ключ $S=(d,n)$ до зашифрованого інформаційного потоку $P(M)$: $S(P(M))=(P(M))^d \bmod n$.

Використання ТЧБ Радемахера в алгоритмі шифрування інформаційних потоків RSA дозволяє зменшити часову складність, яка базується на складності виконання операції $P(M)=M^e \bmod n$. Як показано в літературі, для виконання цієї операції з використанням алгоритму швидкого піднесення до степеня потрібно $O(\ln e)$ операцій множень по модулю. Отже, з врахуванням того, що операція модулярного множення має складність $O(r^2)$, то складність $P(M)=M^e \bmod n$ буде $O(n)=(n^2\cdot \ln e)$, де n - розрядність модуля n. Оскільки в алгоритмі RSA для шифрування та дешифрування використовується одна і та ж операції, то складність цього алгоритму оцінюється, як $O1(n^2\cdot \ln e+2n)$, тоді як запропонованого алгоритму з використанням ТЧБ Радемахера $O3(n)=\begin{cases}\log_2 n\left(3\log_2 n+\dfrac{n}{2}\right), якщо\ n<256\\ n\cdot\left(3\log_2 n+\dfrac{n}{2}\right), в\ інших\ випадках\end{cases}$

(рис. 10.7).

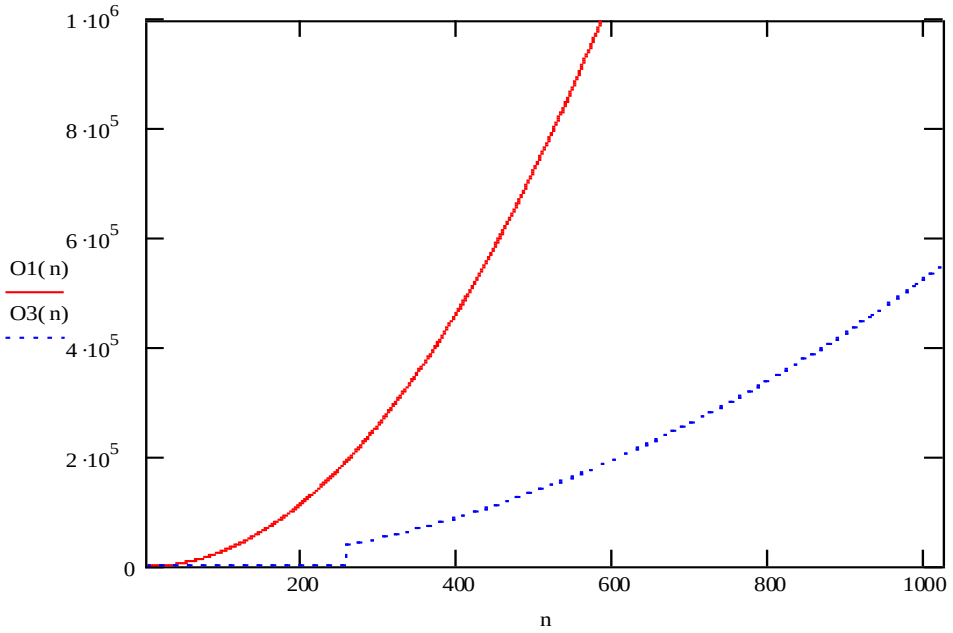


Рис.10.7. Часові складності $O3(n)$ - алгоритму шифрування RSA з використанням розмежованої системи числення Радемахера-Крестенсона, $O1(n)$ - класичного алгоритму RSA.

Основною трудомісткою операцією при шифруванні та дешифруванні інформаційних потоків в алгоритмі RSA є операція модулярного множення та експоненціювання, тому використання запропонованих алгоритмів дозволить значно зменшити складності на 20-40% для параметрів, менших 256 біт, і на 10 % при параметрах від 256 до 1024 біт.

Для реалізації алгоритму шифрування Ель-Гамаля потрібно першочергово згенерувати ключі згідно наступної послідовності дій:

1. Генерується випадкове просте число p довжини n.
2. Вибирається довільне ціле число g, яке є первісним коренем по модулю p, та будь-яке число $x \in (1, p)$ взаємно просте з $p-1$.

Обчислюємо відкритий ключ $y = g^x \bmod p$ з використанням матричного методу піднесення до степеня в базисі Радемахера–Крестенсона. Записуємо в стовпцях матриці величини $g^{2^i} (\bmod n)$ в базисі Радемахера, тобто $g_{ij}=0$, 1 (табл.9.21) та подамо степінь x за степенями 2, тобто $x = x_{n-1}2^{n-1}+\ldots+ +x_i2^i+\ldots+ x_12+x_0$. Шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 10.17. Отже, матричним

методом експоненціювання можна знайти відкритий ключ алгоритму шифрування Ель-Гамаля.

Таблиця 10.21.

Матриця знаходження відкритого ключа алгоритму шифрування Ель -Гамаля в базисі Радемахера–Крестенсона.

$g_{n-1\ n-1}$	…	$g_{i\ n-1}$		$g_{1\ n-1}$	$g_{0\ n-1}$
…	…	…	…	…	…
$g_{n-1\ j}$	…	$g_{i\ j}$	…	$g_{1\ j}$	$g_{0\ j}$
…	…	…	…	…	…
$g_{n-1\ 1}$	…	$g_{i\ 1}$	…	$g_{1\ 1}$	$g_{0\ 1}$
$g_{n-1\ 0}$	…	$g_{i\ 0}$	…	$g_{1\ 0}$	$g_{0\ 0}$
$g^{2^{n-1}}$	…	$g^{2^{i}}$	…	$g^{2^{1}}$	$g^{2^{0}}$

Відкритим ключем в алгоритмі шифрування інформаційних потоків Ель-Гамаля є трійка *(p,g,y)*, закритим – число *x*. Після генерування ключів, шифрування повідомлення M здійснюється згідно алгоритму:

1. Вибирається випадково секретне число k, взаємно просте з p − 1.
2. Обчислюється $a=g^k \bmod p$, $b=y^k M \bmod p$, де M — інформаційний потік який шифрується. Для обчислення $b = y^k M mod\ p$ потрібно послідовно застосувати алгоритм модулярного експоненціювання та модулярного множення в розмежованій системі числення Радемахера – Крестенсона.

Пара чисел *(a,b)* називається шифротекстом, причому довжина шифротексту в алгоритмі шифрування інформаційних потоків Ель-Гамаля вдвоє більша від повідомлення M.

Таким чином, знаючи секретний ключ *x*, вихідне повідомлення можна обчислити з шифротексту *(a,b)* по формулі: $M = b \cdot (a^x)^{-1} \bmod p$.

Стійкість даної системи базується на часовій складності задачі дискретного логарифмування. Як і в алгоритмі шифрування RSA, так і в Ель-Гамаля основною операцією є модулярне експоненціювання. Тому часова складність алгоритму Ель-Гамаля, з врахуванням генерування ключів, буде $O1(n) = (n^2(3n+1))$. Використання розмежованої системи числення Радемахера – Крестенсона дозволяє зменшити складність з $O1(n^2(3n+1))$ до $O3(n) = \begin{cases} \log_2 n \cdot \left(\log_2 n + \frac{n}{2} \log_2 n + 1 \right), якщо n < 256 \\ n \cdot \left(\log_2 n + \frac{n}{2} \log_2 n + 1 \right), 256 \le n \le 1024 \end{cases}$, де n - розрядність модуля p (рис.10.8).

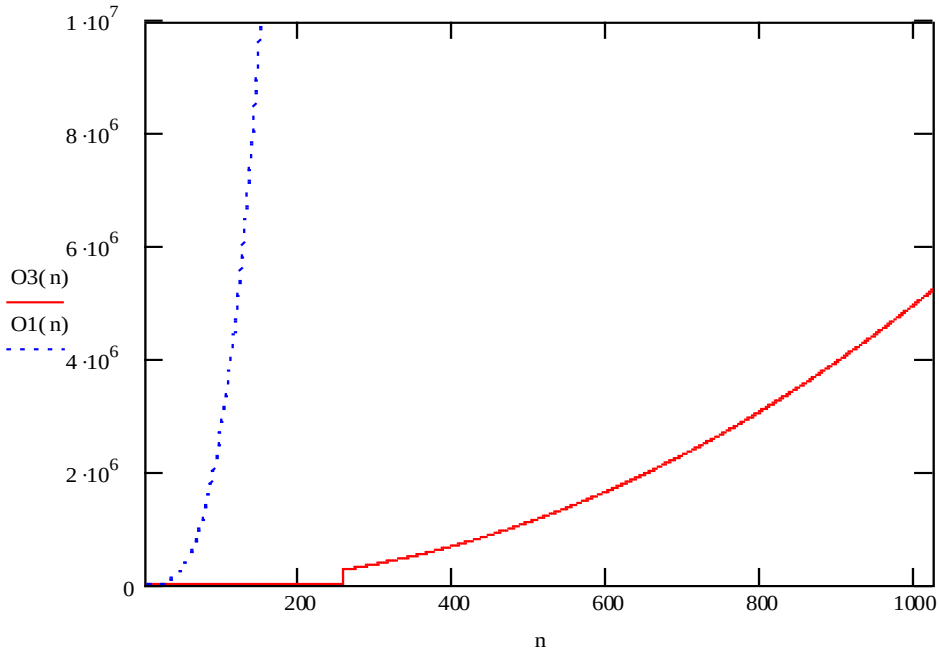


Рис.10.8. Складності $O3(n)$ – алгоритму шифруваня Ель-Гамаля з використанням розмежованої системи числення Радемахера-Крестенсона, $O1(n)$ – класичного алгоритму Ель-Гамаля.

Отже, використання розмежованої системи числення Радемахера – Крестенсона в задачах захисту інформаційних потоків на основі алгоритмів RSA та Ель-Гамаля, дозволяє ефективно застосовувати матричні методи при побудові мультибазисних процесорів шифрування, а заміна операцій множень операціями додавання на етапах генерування ключів, шифрування та дешифрування – дозволяє значно зменшити часову складність від 10 до 40 % в залежності від розрядності параметрів алгоритмів RSA та Ель-Гамаля.

10.5. Метод побудови розподіленого температурного сенсора на основі системи числення базису Крестенсона.

Значна кількість промислових об’єктів України належить до класу стратегічних, автоматизоване управління якими здійснюється засобами спеціалізованих комп’ютерних систем. Їх важливим підкласом є спеціалізовані системи контролю зберігання та обліку руху нафто- та газопродуктів, технологічні об’єкти для яких характеризуються специфічними особливостями, які виключають можливість безпосереднього доступу людини. Однією з головних задач для них є визначення тиску та розподілених температурних полів у різних точках та на різних рівнях. Такі

задачі особливо актуальні також в геофізиці, нафтогазовидобувній, вугільній, космічній та інших галузях промисловості, а також в метеорології.

Сенсори для вимірювання розподілених температурних полів, як правило, реалізуються двома способами: або багатоточковою структурою з паралельним інформаційним каналом, або на основі позиційних систем числення з моноканальною лінією передачі інформації. Недоліком першого способу є наявність великої кількості автономних ліній зв'язку від сенсорів до мікроконтролерів.

В основі другого методу побудови багатопараметричного сенсора температурних полів (БСТП) лежить послідовне з'єднання терморезисторів R_i, опір кожного з яких може змінюватися стрибкоподібно на величину ΔR_i, що визначає точність вимірювання і відповідає основі системи числення. Тоді загальний опір БСТП визначається аналітичним виразом $R_x = \sum_{i=1}^{n} R_i$.

Недоліком даного методу вимірювання температурних полів є значна різниця між ΔR_i, яка відповідає $\frac{\Delta R_{i+1}}{\Delta R_i} = A$, де A – основа позиційної системи числення. Наприклад, при А=2 та А=10 відповідно ΔR_i в кожному каналі повинно змінюватися в 2 та 10 разів. Крім того, кожний i–ий сенсор має в A разів менший діапазон вимірювання температурних полів.

10.5.1. Метод побудови багатопараметричного сенсора температурних полів у системі числення залишкових класів базису Крестенсона.

Нехай маємо n однакових послідовно з'єднаних терморезисторів R_0 (рис.1) , опір яких може змінюватися стрибкоподібно з кроком $\Delta R_i = \frac{R_0}{p_i}$, де p_i–взаємнопрості числа або модулі, які визначають точність вимірювання. При цьому забезпечується діапазон вимірюваних температурних полів $D_i = R_0\left(1 - \frac{1}{p_i}\right)$.

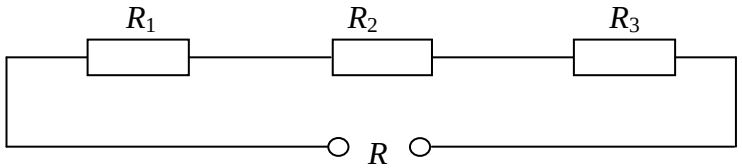


Рис.10.9. Схема вимірювальної системи

Загальний опір $R_x=\sum_{i=1}^{n} R_i$, який визначається безпосереднім вимірюванням, для системи, показаної на рисунку 10.9, потрібно представити у вигляді суми шуканих опорів R_i. Знайдемо діапазон вимірювання $P=\prod_{i=1}^{n} p_i$ та базисні числа m_i. Далі виконується така послідовність дій: $X=R_x mod R_0$, $Y=\frac{XP}{R_0}$, $b_i=Y mod\ p_i$. Шукані опори кожного резистора визначаються з виразу $R_i=(b_i m_i \Delta R_i) mod\ p_i$.

Для прикладу візьмемо R_0=100 Ом, p_1=99, p_2=100, p_3=101. Тоді P=999900; ΔR_1=1,01 Ом; ΔR_2=1 Ом; ΔR_3=0,99 Ом; m_1=50, m_2=99, m_3=51. Нехай вимірювальний прилад зафіксував величину R_x=249,3166 Ом. Як наслідок отримуємо X=249,3166*mod*100=49,3166; $Y=\frac{49{,}3166\cdot 999900}{100}=493117$; b_1=493117*mod* 99=97; b_2=493117*mod* 100=17; b_3=493117*mod* 101=35. Шукані величини $R_1 \approx (97\cdot 50\cdot 1{,}01)$ mod99=98,98; $R_2 \approx (17\cdot 99\cdot 1)$ mod100=83; $R_3 \approx (35\cdot 51\cdot 0{,}99)$ mod101=67,32.

10.5.2. Метод побудови багатопараметричного сенсора температурних полів за допомогою таблиць.

Шукані опори R_i можна шукати за допомогою таблиць. Нехай R_0=10 Ом, p_1=3, p_2=4, p_3=5. Тоді P=60; ΔR_1=3,33 Ом; ΔR_2=2,5 Ом; ΔR_3=2 Ом. Будуємо таблицю 10.22, для якої введемо позначення: $R_x^{(1)}=R_x\ mod\ R_0$, причому значення $R_x^{(1)}$ розміщені в таблиці 1 в порядку їх зростання; $R_x^{(2)}$, $b_1^{(1)}$, $b_2^{(1)}$, $b_3^{(1)}$ – значення R_x, b_1, b_2, b_3, які відповідають $R_x^{(1)}$.

Якщо вимірювальний прилад зафіксував R_x=12,83 Ом, то $R_x^{(1)}=12{,}83\ mod\ 10=2{,}83$. Шукаємо дане значення в таблиці і йому відповідають $b_1^{(1)}$=1, $b_2^{(1)}$=3, $b_3^{(1)}$=1. Шукані значення R_1=1·3,33=3,33 Ом; R_2=3·2,5=7,5 Ом; R_3=1·2=2 Ом.

Таблиця 10.22.

Побудова БСТП табличним методом для p_1=3, p_2=4, p_3=5.

N п/п	b_1	b_2	b_3	R_x	$R_x^{(1)}$	$R_x^{(2)}$	$b_1^{(1)}$	$b_2^{(1)}$	$b_3^{(1)}$
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0

продовження таблиці 10.22.

1	1	1	1	7,83	0,17	20,17	2	3	3
2	2	2	2	15,67	0,33	10,33	1	2	1
3	0	3	3	13,5	0,5	10,5	0	1	4
4	1	0	4	11,33	0,67	10,67	2	0	2
5	2	1	0	9,17	0,83	10,83	1	3	0
6	0	2	1	7	1	11	0	2	3
7	1	3	2	14,83	1,17	11,17	2	1	1
8	2	0	3	12,67	1,33	11,33	1	0	4
9	0	1	4	10,5	1,5	11,5	0	3	2
10	1	2	0	8,33	1,67	11,67	2	2	0
11	2	3	1	16,17	1,83	11,83	1	1	3
12	0	0	2	4	2	2	0	0	1
13	1	1	3	11,83	2,17	22,17	2	3	4
14	2	2	4	19,67	2,33	12,33	1	2	2
15	0	3	0	7,5	2,5	2,5	0	1	0
16	1	0	1	5,33	2,67	12,67	2	0	3
17	2	1	2	13,17	2,83	12,83	1	3	1
18	0	2	3	11	3	13	0	2	4
19	1	3	4	18,83	3,17	13,17	2	1	2
20	2	0	0	6,67	3,33	13,33	1	0	0
21	0	1	1	4,5	3,5	13,5	0	3	3
22	1	2	2	12,33	3,67	13,67	2	2	1
23	2	3	3	20,17	3,83	13,83	1	1	4
24	0	0	4	8	4	4	0	0	2
25	1	1	0	5,83	4,17	14,17	2	3	0
26	2	2	1	13,67	4,33	14,33	1	2	3
27	0	3	2	11,5	4,5	4,5	0	1	1
28	1	0	3	9,33	4,67	14,67	2	0	4
29	2	1	4	17,17	4,83	14,83	1	3	2
30	0	2	0	5	5	5	0	2	0
31	1	3	1	12,83	5,17	15,17	2	1	3
32	2	0	2	10,67	5,33	5,33	1	0	1
33	0	1	3	8,5	5,5	15,5	0	3	4
34	1	2	4	16,33	5,67	15,67	2	2	2
35	2	3	0	14,17	5,83	5,83	1	1	0
36	0	0	1	2	6	6	0	0	3
37	1	1	2	9,83	6,17	16,17	2	3	1
38	2	2	3	17,67	6,33	16,33	1	2	4
39	0	3	4	15,5	6,5	6,5	0	1	2
40	1	0	0	3,33	6,67	6,67	2	0	0
41	2	1	1	11,17	6,83	16,83	1	3	3

продовження таблиці 10.22.

42	0	2	2	9	7	7	0	2	1
43	1	3	3	16,83	7,17	17,17	2	1	4
44	2	0	4	14,67	7,33	7,33	1	0	2
45	0	1	0	2,5	7,5	7,5	0	3	0
46	1	2	1	10,33	7,67	17,67	2	2	3
47	2	3	2	18,17	7,83	7,83	1	1	1
48	0	0	3	6	8	8	0	0	4
49	1	1	4	13,83	8,17	18,17	2	3	2
50	2	2	0	11,67	8,33	8,33	1	2	0
51	0	3	1	9,5	8,5	8,5	0	1	3
52	1	0	2	7,33	8,67	8,67	2	0	1
53	2	1	3	15,17	8,83	18,83	1	3	4
54	0	2	4	13	9	9	0	2	2
55	1	3	0	10,83	9,17	9,17	2	1	0
56	2	0	1	8,67	9,33	9,33	1	0	3
57	0	1	2	6,5	9,5	9,5	0	3	1
58	1	2	3	14,33	9,67	19,67	2	2	4
59	2	3	4	22,17	9,83	9,83	1	1	2

10.5.3. Рекомендації щодо вибору наборів модулів для вимірювання температурних полів з точки зору теорії чисел.

Для досягнення приблизно однакової точності вимірювання опору кожного резистора вибрані взаємно прості модулі не повинні відрізнятися дуже сильно. Прикладом може бути розглянутий вище, набір модулів з трьох послідовних чисел p_1=99, p_2=100, p_3=101 для R_0=100 Ом. За умови більшої кількості модулів їх можна вибрати таким чином: p_1=101, p_2=102, p_3=103, p_4=107, p_5=109, … .

З точки зору теорії чисел для істотного зменшення кількості обчислень модулі потрібно вибрати так, щоб вони утворювали досконалу форму системи залишкових класів базису Крестенсона, для яких m_i=1. У таблиці 10.23 представлено можливі значення відповідних параметрів при p_1=2, p_2=3, p_3=5, R_0=10 Ом, ΔR_1=5 Ом; ΔR_2=3,33 Ом; ΔR_3=2 Ом. Нехай прилад зафіксував R_x=13,67 Ом. Тоді X=13,67mod10=3,67; $Y = \frac{3{,}67 \cdot 30}{10} = 11$; b_1=11mod 2=1; b_2=11mod 3=2; b_3=11mod 5=1. Шукані величини R_1 =1·5=5 Ом; R_2=2·3,33=6,67 Ом; R_3 =1·2=2 Ом.

Недоліком даного методу є те, що в досконалій формі базису Крестенсона модулі дуже швидко зростають і, відповідно, істотно відрізняються точності вимірювання температурних полів у різних точках.

Якщо температури необхідно виміряти лише у двох точках, то набір модулів зручно вибрати у вигляді двох великих послідовних чисел, що забезпечує приблизно однакову високу точність. У таблиці 10.24 представлено можливі значення відповідних параметрів при p_1=5, p_2=6, R_0=10 Ом, ΔR_1=2 Ом; ΔR_2=1,67 Ом. Тоді m_1=1, m_2=5*mod* 6=–1 *mod* 6. Це означає, що $b_1^{(1)}=b_1$, $b_2^{(1)}=-b_2$ *mod* 6=6– b_2 *mod* 6.

Нехай, наприклад, прилад зафіксував величину R_x=12,67 Ом. За допомогою таблиці 10.24 можна знайти, що R_1 =3·2=6 Ом; R_2=4·1,67=6,67 Ом. Аналітичним методом будемо мати: X=12,67*mod*10=2,67; $Y=\frac{2{,}67\cdot 30}{10}=8$; b_1=8*mod* 5=3; b_2=8*mod* 6=2. Шукані значення R_1 =3·2=6 Ом; R_2=(6–2)·1,67=6,67 Ом.

Таблиця 10.23.
Побудова БСТП табличним методом для p_1=2, p_2=3, p_3=5.

N п/п	b_1	b_2	b_3	R_x	$R_x^{(1)}$
0	0	0	0	0	0
1	1	1	1	10,33	0,33
2	0	2	2	10,67	0,67
3	1	0	3	11	1
4	0	1	4	11,33	1,33
5	1	2	0	11,67	1,67
6	0	0	1	12	2
7	1	1	2	12,33	2,33
8	0	2	3	12,67	2,67
9	1	0	4	13	3
10	0	1	0	13,33	3,33
11	1	2	1	13,67	3,67
12	0	0	2	14	4
13	1	1	3	14,33	4,33
14	0	2	4	14,67	4,67
15	1	0	0	15	5
16	0	1	1	15,33	5,33
17	1	2	2	15,67	5,67
18	0	0	3	16	6
19	1	1	4	16,33	6,33
20	0	2	0	16,67	6,67

Таблиця 10.24.
Побудова БСТП табличним методом для p_1=5, p_2=6.

N п/п	b_1	b_2	R_x	$R_x^{(1)}$	$R_x^{(2)}$	$b_1^{(1)}$	$b_2^{(1)}$
0	0	0	0	0	0	0	0
1	1	1	3,67	0,33	10,33	1	5
2	2	2	7,33	0,67	10,67	2	4
3	3	3	11	1	11	3	3
4	4	4	14,67	1,33	11,33	4	2
5	0	5	8,33	1,67	1,67	0	1
6	1	0	2	2	2	1	0
7	2	1	5,67	2,33	12,33	2	5
8	3	2	9,33	2,67	12,67	3	4
9	4	3	13	3	13	4	3
10	0	4	6,67	3,33	3,33	0	2
11	1	5	10,33	3,67	3,67	1	1
12	2	0	4	4	4	2	0
13	3	1	7,67	4,33	14,33	3	5
14	4	2	11,33	4,67	14,67	4	4
15	0	3	5	5	5	0	3
16	1	4	8,67	5,33	5,33	1	2
17	2	5	12,33	5,67	5,67	2	1
18	3	0	6	6	6	3	0
19	4	1	9,67	6,33	16,33	4	5
20	0	2	3,33	6,67	6,67	0	4

продовження таблиці 10.24.

21	1	0	1	17	7
22	0	1	2	17,33	7,33
23	1	2	3	17,67	7,67
24	0	0	4	18	8
25	1	1	0	18,33	8,33
26	0	2	1	18,67	8,67
27	1	0	2	19	9
28	0	1	3	19,33	9,33
29	1	2	4	19,67	9,67

21	1	3	7	7	7	1	3
22	2	4	10,67	7,33	7,33	2	2
23	3	5	14,33	7,67	7,67	3	1
24	4	0	8	8	8	4	0
25	0	1	1,67	8,33	8,33	0	5
26	1	2	5,33	8,67	8,67	1	4
27	2	3	9	9	9	2	3
28	3	4	12,67	9,33	9,33	3	2
29	4	5	16,33	9,67	9,67	4	1

Виходячи з вищесказаного, можна зробити висновок, що використання системи залишкових класів базису Крестенсона дозволяє ефективно визначати опір кожного із декількох послідовно з’єднаних резисторів за відомим їх сумарним опором.

РОЗДІЛ 11

ФОРМУВАННЯ ІНФОРМАЦІЙНИХ ПОВІДОМЛЕНЬ У КОДАХ ПОЛЯ ГАЛУА

11.1. Формування цифрових повідомлень на основі КПГ.

Для генерування кодів Галуа використовуються незвідні алгебраїчні поліноми, які породжують ключі генерування кодів поля Галуа (табл.11.1).

Таблиця 11.1.

Таблиця ключів кодів Галуа.

Розрядність коду	Ключ коду Галуа
4	$x_1 \oplus x_4$
5	$x_2 \oplus x_5$
6	$x_1 \oplus x_6$
7	$x_3 \oplus x_7$
8	$x_3 \oplus x_8$
9	$x_4 \oplus x_9$
10	$x_3 \oplus x_{10}$
11	$x_2 \oplus x_{11}$
12	$x_1 \oplus x_4 \oplus x_6 \oplus x_{12}$
13	$x_1 \oplus x_3 \oplus x_4 \oplus x_{13}$
14	$x_1 \oplus x_6 \oplus x_{10} \oplus x_{14}$
15	$x_1 \oplus x_{15}$
16	$x_1 \oplus x_3 \oplus x_{12} \oplus x_{16}$
17	$x_3 \oplus x_{17}$
18	$x_7 \oplus x_{18}$
19	$x_1 \oplus x_2 \oplus x_5 \oplus x_{19}$
20	$x_1 \oplus x_{18}$
21	$x_2 \oplus x_{21}$
22	$x_1 \oplus x_{22}$
23	$x_5 \oplus x_{23}$
24	$x_1 \oplus x_2 \oplus x_7 \oplus x_{24}$
25	$x_1 \oplus x_4$
26	$x_1 \oplus x_2 \oplus x_3 \oplus x_7$
27	$x_1 \oplus x_2 \oplus x_3 \oplus x_6$
28	$x_1 \oplus x_4$
29	$x_1 \oplus x_3$
30	$x_1 \oplus x_2 \oplus x_5 \oplus x_7$
31	$x_1 \oplus x_7$
32	$x_1 \oplus x_3 \oplus x_7 \oplus x_8$

Важливою перевагою кодової послідовності Галуа (КПГ) є їх проста генерація на основі рекурентного рівняння. Найпростіші ключі кодів Галуа описуються виразом:

$$G_{i+1} = G_i \oplus G_{i-1},$$

де G_i - біти коду Галуа.

При *n=4* отримаємо код (рис.11.1).

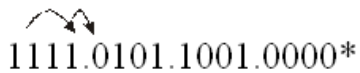


Рис.11.1. Формування коду Галуа при n=4 (* - біт-стаффінг).

Коди поля Галуа належать до класу рекурентних кодів, які широко використовуються для захисту інформації від несанкціонованого доступу згідно структури кодування байтів, приведеної на рис.11.2.

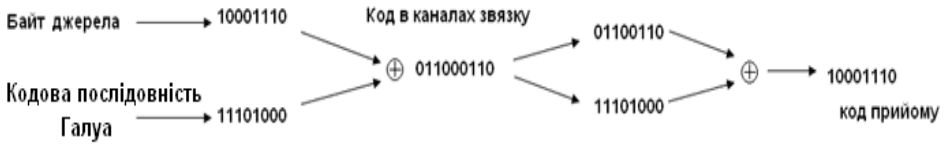


Рис.11.2. Байт, який передається на основі кодової послідовності Галуа.

В загальному випадку генератор коду поля Галуа реалізується на регістрі зсуву зі зворотніми зв'язками, згідно коефіцієнтів незвідного полінома (рис.11.3).

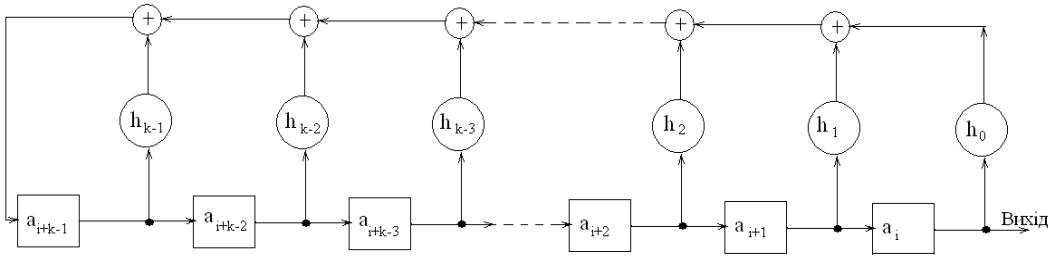


Рис.11.3. Генератор коду Галуа на регістрі зсуву зі зворотнім зв'язком.

Кодову послідовність Галуа можна представити у вигляді рекурентного кільця (рис.11.4).

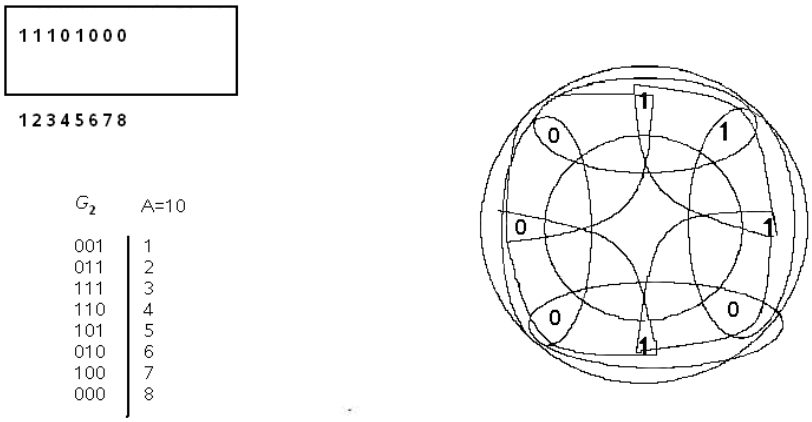


Рис. 11.4. Представлення кодової послідовності Галуа у вигляді рекурентного кільця.

На рис.11.5-11.8 представлено структурні схеми генераторів коду поля Галуа.

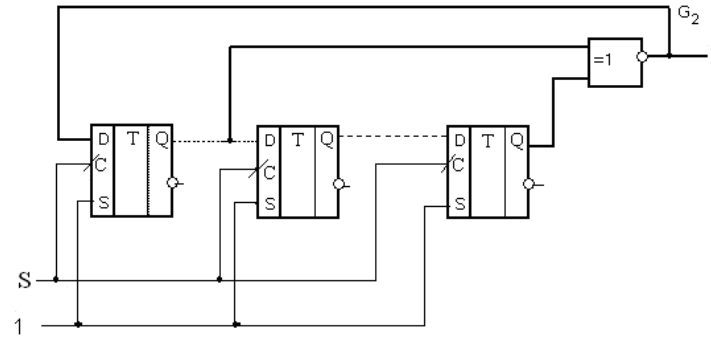


Рис.11.5. Структурна схема генератора коду поля Галуа.

Складність виконання обчислень, а відповідно конструкція і вартість обладнання, які здійснюють ці обчислення, залежать від вибору представлення поля $GF(_p^r)$, де r - степінь незвідних поліномів кінцевого поля Галуа $GF(_p^r)$, (рис.11.6).

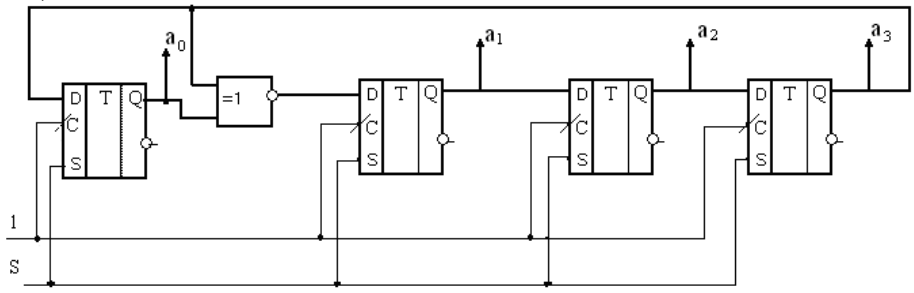


Рис.11.6. Формувач елементів поля Галуа $GF(_2^4)$.

Схему формування елементів поля Галуа $GF(_2^4)$ в зворотному порядку отримаємо, змінивши підключення суматора по *mod2* (рис.11.7).

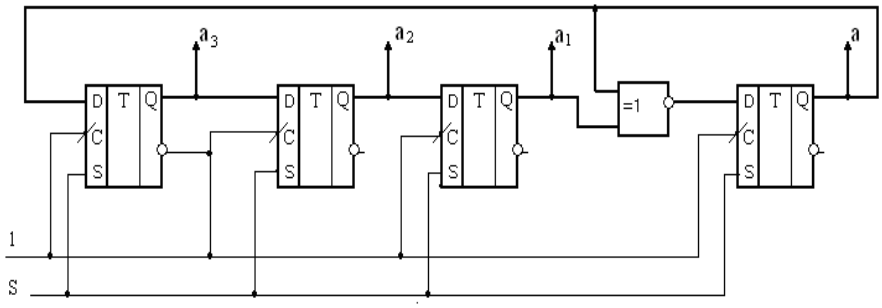


Рис.11.7. Схема формування елементів поля Галуа $GF(_2^4)$ в зворотному порядку.

Ненульові елементи поля утворюють циклічну групу порядку $2^r–1$.

На рис. 11.8 представлено структурну схему генератора коду поля Галуа, в якому відсутні логічні елементи «виключаюче АБО», а логічний зворотній зв'язок реалізується на основі *T*-тригера. Такі генератори КПГ характеризуються простішою схемотехнічною структурою і швидкодією, що перевищує в два рази структури, які включають в зворотному зв'язку додатковий логічний елемент, затримка сигналів якого становить 2τ.

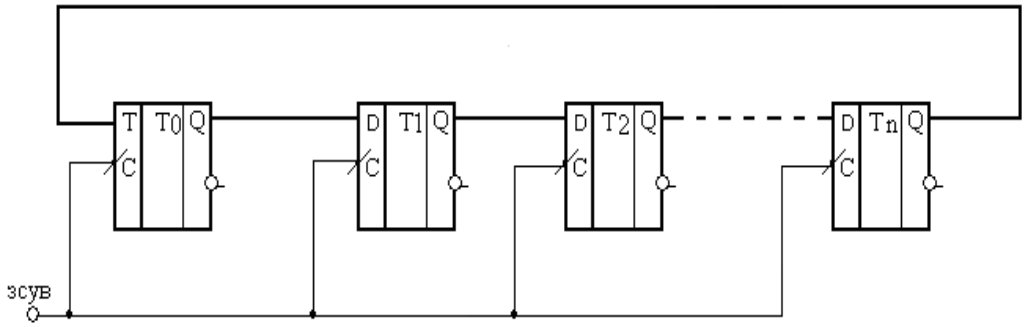


Рис.3.18. Структура генератора з лічильним тригером в ланцюгу зворотного зв'язку.

Функціональними обмеженнями досліджених структур генераторів КПГ є можливість генерування кодової послідовності Галуа $N=2^k–1$, тобто відсутній 2^k нульовий біт Галуа. Тому для генерування сигнальних коректуючих кодів розроблена структура генератора КПГ, в якій відсутнє вказане обмеження (рис.11.9).

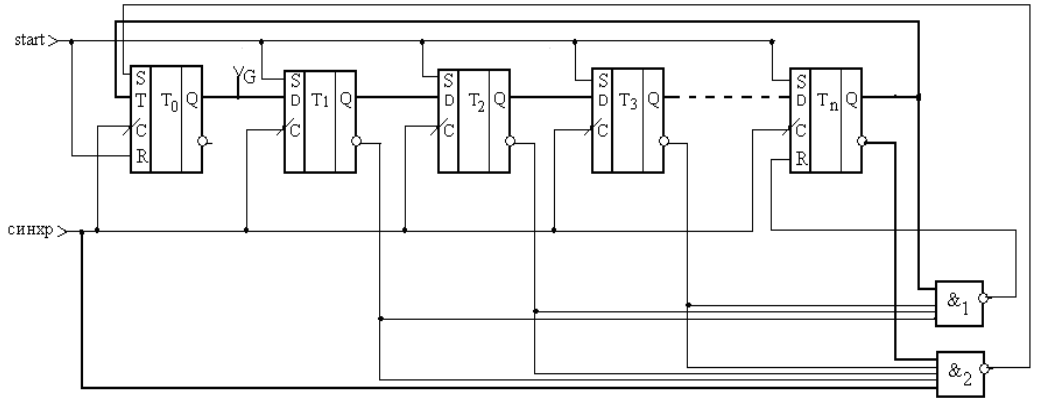


Рис.11.9. Структурна схема генератора Галуа з періодом $N=2^k$.

Даний генератор складається з n *D*-тригерів (T_1, T_2, …T_n), T_0-тригера і двох логічних елементів «*І-НЕ*».

Робота генератора починається формуванням сигналу “start” , який по *S*-входах встановлює всі *D*-тригери в стан «1» , а *T*-тригер по *R*-вході в стан «нуль», що відповідає початковому формуванню Галуа ознак сигнальних кодів після кожного синхросигналу, який подається на всі С-входи всіх тригерів генератора. На виході T_n- тригера формується біт-орієнтована псевдовипадкова послідовність.

Зворотній логічний зв'язок, який формує рекурентну послідовність бітів Галуа реалізується з’єднанням прямого виходу T_n- тригера з *T*-входом T_0-тригера.

Робота такого генератора псевдовипадкової послідовності (рис.11.9) на прикладі $GF\binom{4}{2}$ демонструється кодовою таблицею *G* (табл.11.2).

Таблиця 11.2.

Кодова таблиця генератора псевдовипадкової послідовності.

N/п	T_0	T_1	T_2	T_3	T_4
0	0	1	1	1	1
1	1	0	1	1	1
2	0	1	0	1	1
3	1	0	1	0	1
4	0	1	0	1	0
5	0	0	1	0	1
6	1	0	0	1	0
7	1	1	0	0	1
8	0	1	1	0	0
9	0	0	1	1	0
10	0	0	0	1	1
11	1	0	0	0	1
12	0	1	0	0	0
13	0	0	1	0	0
14	0	0	0	1	0
15	0	0	0	0	1
16	1	0	0	0	0
17	1	1	0	0	0
18	1	1	1	0	0
19	1	1	1	1	0
20	1	1	1	1	1

При досягненні коду Галуа, який відповідає $2^k–1$ комбінації, тобто (000…1) прямий вихід T_n-ного тригера та інверсні виходи всіх інших *D*-тригерів, подані на перший логічний елемент («*І-НЕ*»), на його інверсному виході формують нульовий сигнал, який по *R*-входу скидає *T*-тригер в нуль,

тобто формується 2^n-комбінація коду Галуа (000…0). Наступний синхросигнал, який подається на один з входів другого логічного елемента («*І-НЕ*»), а також інверсні виходи всіх *D*-тригерів на його виході формують нульовий сигнал, який по *S*-входу встановлює T_0- тригер в стан одиниці, що дозволяє генератору завершити цикл формування кодів поля Галуа до стартового положення (111…1).

Аналіз аналітичних виразів незвідних алгебраїчних поліномів показує, що в зворотньому зв'язку може використовуватися парне число логічних коефіцієнтів k=2,4,6,… Отже, швидкодія генераторів КПГ буде знижуватися пропорційно числу послідовно включених логічних елементів «виключаюче АБО» (рис.11.10).

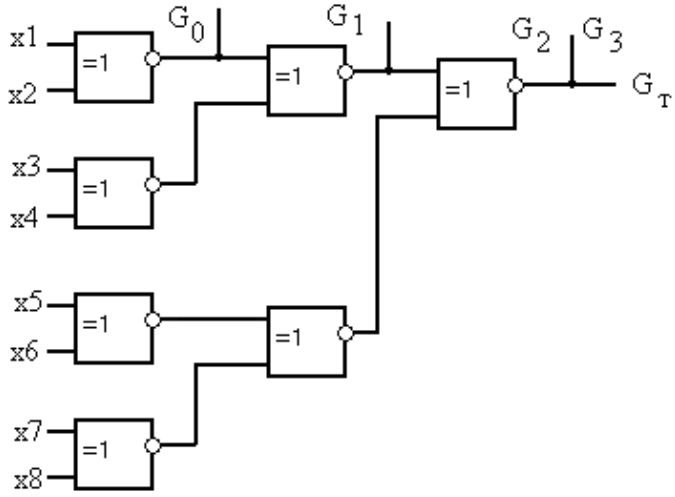


Рис.11.10. Схема логічних зв'язків генератора КПГ при взаємодії різного числа коефіцієнтів незвідного алгебраїчного полінома.

Відповідно, час затримки сигналів в різних сигналах різних типів генераторів КПГ буде описуватись наступними рівняннями:

$\tau_0 = T_T + T_D$; $\tau_1 = 2\tau + T_D$; $\tau_2 = 4\tau + T_D$; $\tau_3 = 6\tau + T_D$; $\tau_4 = T_T + T_D + V$;

де $T_T = V$, $T_D = V$, τ ("=1") $= 2V$;

V – час затримки одного вентиля (транзистора) мікроелектроніки.

Аналіз цих рівнянь показує, що швидкодія досліджуваних структурних схем генераторів КПГ не залежить від числа *D*-тригерів, оскільки вони спрацьовують синхронно за 1 такт, тобто швидкодія таких генераторів не залежить від довжини генерованого коду поля Галуа, що показано на рис.11.11.

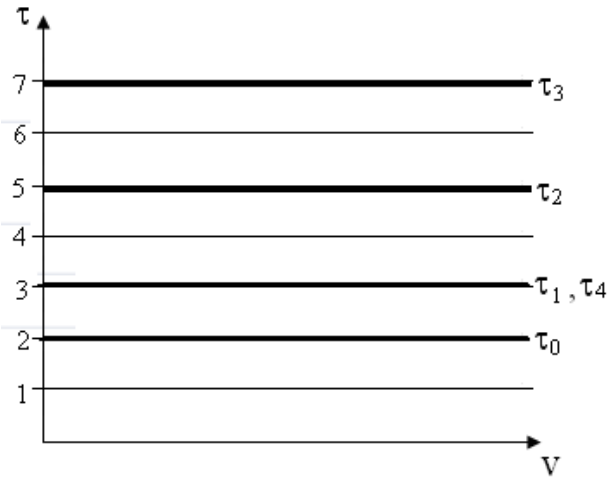


Рис. 11.11. Оцінка часової складності різних структурних схем генераторів КПГ.

Аналіз рис. 11.11 показує, що генератори КПГ, в яких відсутні логічні елементи «виключаюче АБО», характеризуються максимальною швидкодією.

В той же час генератори КПГ, що використовують поліноми з максимальним числом коефіцієнтів, які використовуються для організації зворотнього логічного зв'язку, характеризуються більш високими коректуючими властивостями, при їх використанні в сигнальних кодах, по відношенню до неприводимих поліномів з мінімальним числом таких коефіцієнтів.

11.2. АЦП на основі кодів поля Галуа.

В результаті аналого-цифрового перетворення можуть бути отримані дані в інших дискретних базисах, які дозволяють усунути цей недолік при використанні відповідних АЦП. Прикладом можуть служити скануючі Галуа-перетворювачі та побудований на їх базі АЦП, зображений на рис.11.12.

На виході таких АЦП формується послідовний код Галуа, який однозначно відповідає аналоговому сигналу на вході. Використання АЦП Галуа має певні переваги над класичними перетворювачами з паралельним двійковим кодом на виході, які зумовлюються перевагами кодів поля Галуа і можливістю спростити схему аналого-цифрового перетворювача.

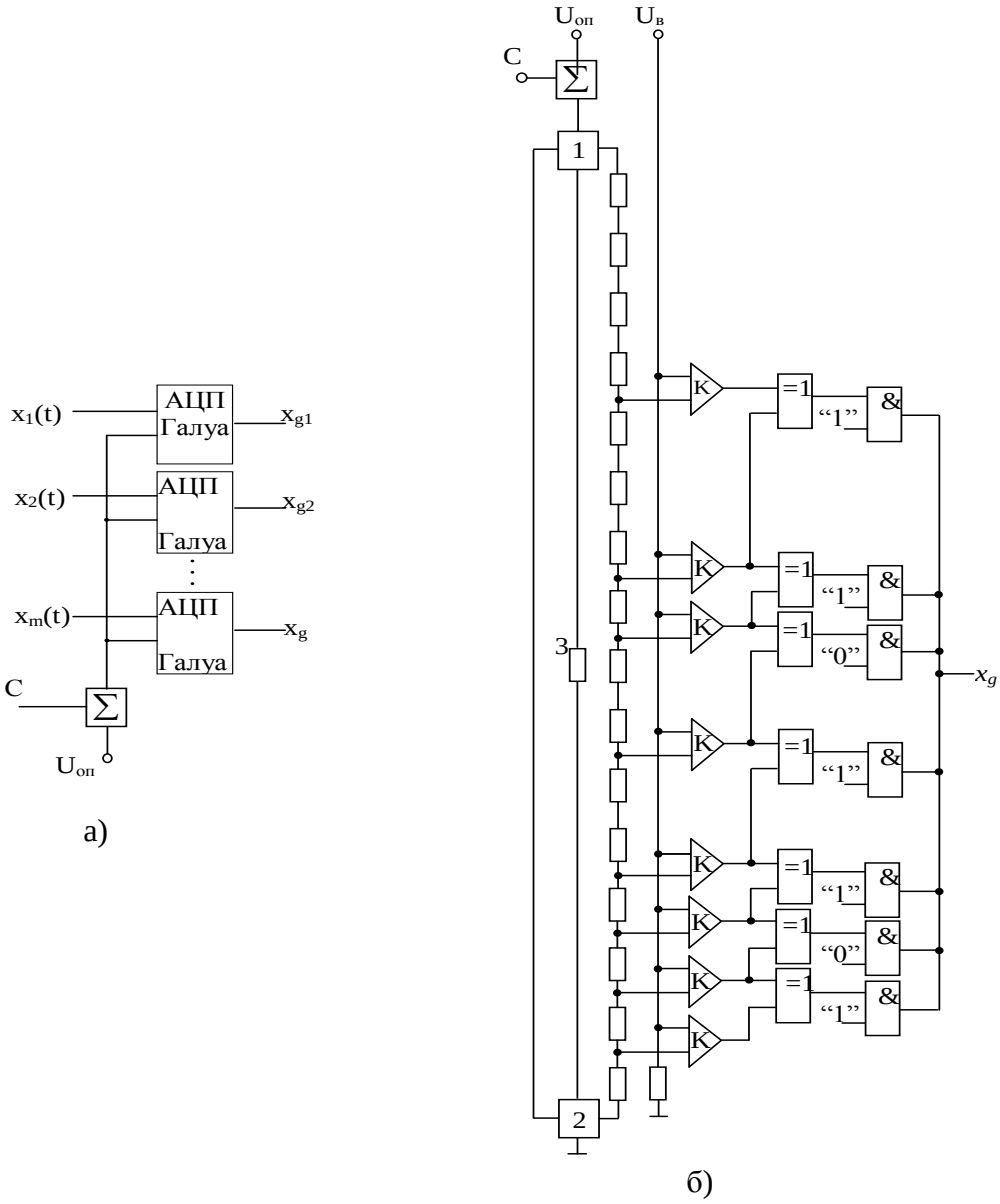


Рис.11.12. Схема модифікованого
а) – багатоканального; б) – одноканального АЦП Галуа.
1, 2 – багатостабільні елементи; 3 – струмообмежуючий резистор;
$\sum$ – суматор; К – компаратор; x_g – вихідний код Галуа.

Галуа-перетворювачі мають регулярну структуру, за рахунок чого є

простими в реалізації на сучасних програмованих логічних матрицях. Генерація на виході послідовного коду Галуа дає можливість паралельного зчитування інформації з усіх каналів, а також забезпечує завадозахищеність та можливість виправлення помилок шляхом введення додаткових бітів Галуа.

Проте швидкодію АЦП Галуа знижує те, що на виході формується послідовний код, для чого потрібно n тактових імпульсів, n – розрядність вихідного коду.

Модифікована архітектура АЦП розрядністю p (рис.11.12), має більш просту схему, в порівнянні з описаною, в якій використовується в два рази менше найбільш складних компараторних елементів, тобто $N/2$, при $N = 2^p$. Зменшення числа цих елементів у порівнянні з класичними АЦП паралельного типу досягається за рахунок відсутності ланок, які відповідають одиничним значенням коду поля Галуа при їх повтореннях.

На рис.11.13. наведено діаграми аналого-цифрового перетворення розглянутих типів багатоканальних АЦП скануючого типу у базисі Галуа.

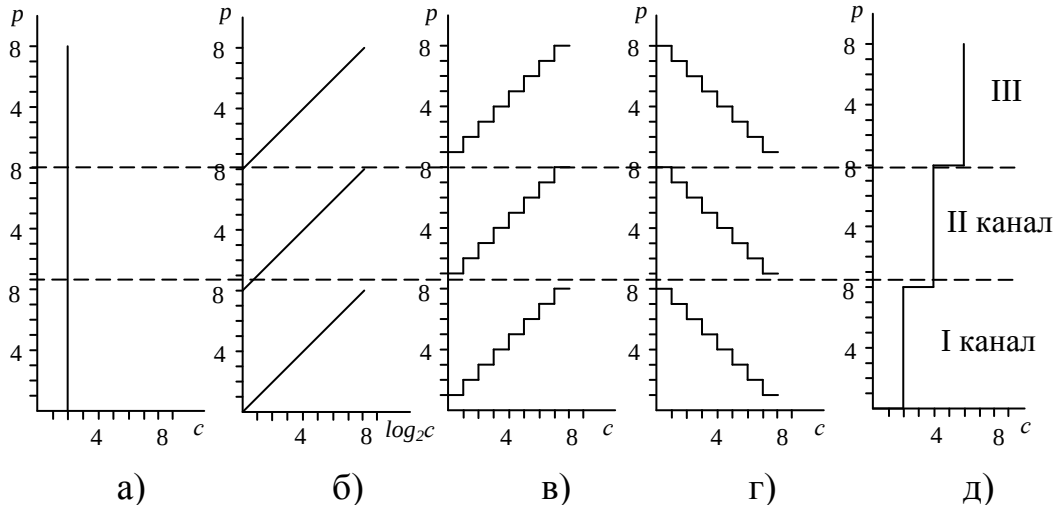


Рис.11.13. Діаграми аналого-цифрового перетворення даних різними типами АЦП: а) – паралельного; б) – з паралельною розгорткою; в) – АЦП Галуа; г) – порозрядного наближення; д) – послідовного АЦП з комутатором каналів. C – кількість тактів АЦП; p – розряди цифрових відліків.

В табл.11.3 приведено приклад перетворення інформації в АЦП такого класу при $p = 4$, $N = 16$, $U_{вх} = 9$ і $U_{вх} = 13$. Номери каналів, в яких відсутні ланки елементів позначені “*”.

Таблиця 11.3.

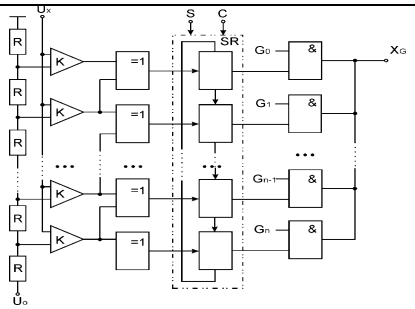
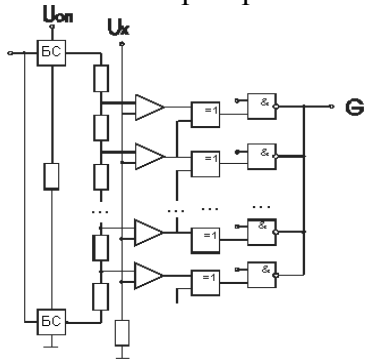
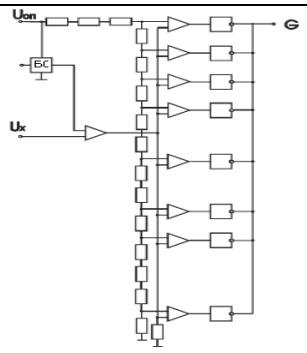
Проміжні та вихідні коди, що формуються елементами перетворювача Галуа при вхідних аналогових сигналах $U_{вх} = 9$ і $U_{вх} = 13$

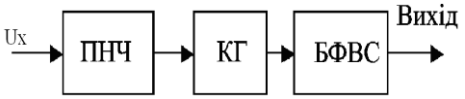
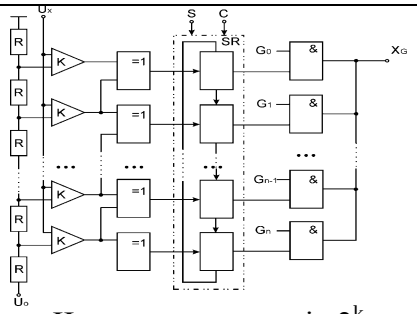
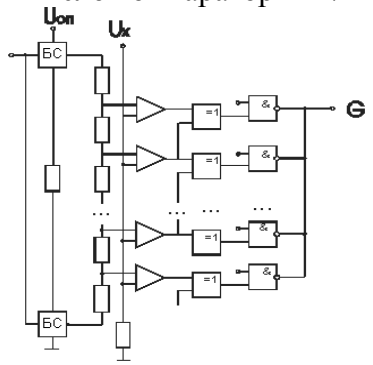
№ каналу	Коди на виході компараторів		Коди на входах елементів «І»			Код Галуа на виході перетво-рювача
			вхід 1		вхід 2	
	$U_{вх}$=13	$U_{вх}$=9	$U_{вх}$=13	$U_{вх}$=9		
15*	1	0	1	0	0	
14*	—	—	—	—	—	
13*	—	—	—	—	—	1110
12	1	1	0	1	1	
11*	—	—	—	—	—	
10*	—	—	—	—	—	
9	—	—	—	—	—	1101
8	1	1	0	0	0	
7*	1	1	0	0	1	
6	—	—	—	—	—	
5*	1	1	0	0	0	
4	—	—	—	—	—	
3	1	1	0	0	1	
2	1	1	0	0	0	
1	1	1	0	0	1	
0*	—	—	—	—	—	

На рис.11.13 наведено діаграми аналого-цифрового перетворення розглянутих типів одноканальних АЦП скануючого типу у базисі Галуа.

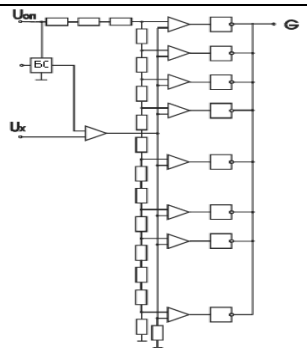
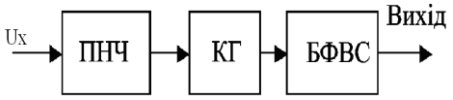
Таблиця 11.4.

Структурні схеми АЦП у базисі Галуа

Тип АЦП	Структура	Параметри вихідних сигналів
1	2	3
Паралельний в базисі Галуа з буферним регістром	 Число компараторів 2k	Послідовний код Галуа, gk gk-1 … gv gv-1 … g1
Скануючий в базисі Галуа	Число компараторів 2k/2 	Послідовний код Галуа, gk gk-1 … gv gv-1 … g1
Скануючий в базисі Галуа на імпульсних компараторах	 k=4 Число компараторів $2^k/2$	Послідовний код Галуа, g_k g_{k-1} … g_v g_{v-1} … g_1

1	2	3
Інтегрально-імпульсний АЦП з імпульсним вихідним кодом в базисі Крейга-Галуа		Послідовний код в базисі Галуа … g_k … g_v g_{v-1} … g_1 …
Паралельний в базисі Галуа з буферним регістром	 Число компараторів 2^k	Послідовний код Галуа, g_k g_{k-1} … g_v g_{v-1} … g_1
Скануючий в базисі Галуа	Число компараторів $2^k/2$ 	Послідовний код Галуа, g_k g_{k-1} … g_v g_{v-1} … g_1

продовження таблиця 11.4.

1	2	3
Скануючий в базисі Галуа на імпульсних компараторах	 k=4 Число компараторів $2^k/2$	Послідовний код Галуа, g_k g_{k-1} … g_v g_{v-1} … g_1
Інтегрально-імпульсний АЦП з імпульсним вихідним кодом в базисі Крейга-Галуа		Послідовний код в базисі Галуа … g_k … g_v g_{v-1} … g_1 …

11.3. .Архітектури та характеристики багатоканальних АЦП Галуа.

Як видно з аналізу характеристик АЦП Галуа, які належать до класу кращих архітектур серед відомих АЦП, особливі позитивні ознаки, а саме, наявність половини компараторів в порівнянні з паралельними АЦП при однаковій розрядності, а також відсутність пірамідального дешифратора, мають АЦП Галуа на основі імпульсних компараторів, виходи яких через інвертори формують на вихідній шині дані на основі провідного елемента “АБО”.

Недоліком даної архітектури є структурна надлишковість, а також можливий дрейф вхідної напруги в процесі її сканування та формування послідовного *k* розрядного кодону Галуа. На рис.11.14 показана вдосконалена архітектура АЦП Галуа на основі імпульсних компараторів з інверсним виходом, пристроєм вибірки зберігання та захистом вихідних даних від помилок (на прикладі *k*=4).

Перевагою даного АЦП є наявність половини компараторів в порівнянні з аналогічними АЦП паралельного типу, де кількість компараторів рівна 2^k. В даному випадку використовується 2 додаткові біти захисту від помилок(n=2), так як в застосовуваному коді Галуа, який як відомо має рекурентні властивості, наступні біти Галуа мають послідовність 00…01 то використовується 1 додатковий компаратор, так як компаратори ставляться лише в позиціях де значення коду рівне 1.

Даний АЦП складається з пристрою вибірки і зберігання ПВЗ, багатостабільного елемента БС, суматора, лінійки резисторів R на яку подається опорна напруга U_0 та лінійки імпульсних компараторів K, які спрацьовують при зміні знаку різницевого сигналу на прямому та інвертованому входах компаратора. Використання інвертованого виходу компаратора типу відкритий колектор дозволяє об'єднати їх в спільну шину, з якої знімається вихідний G код даних.

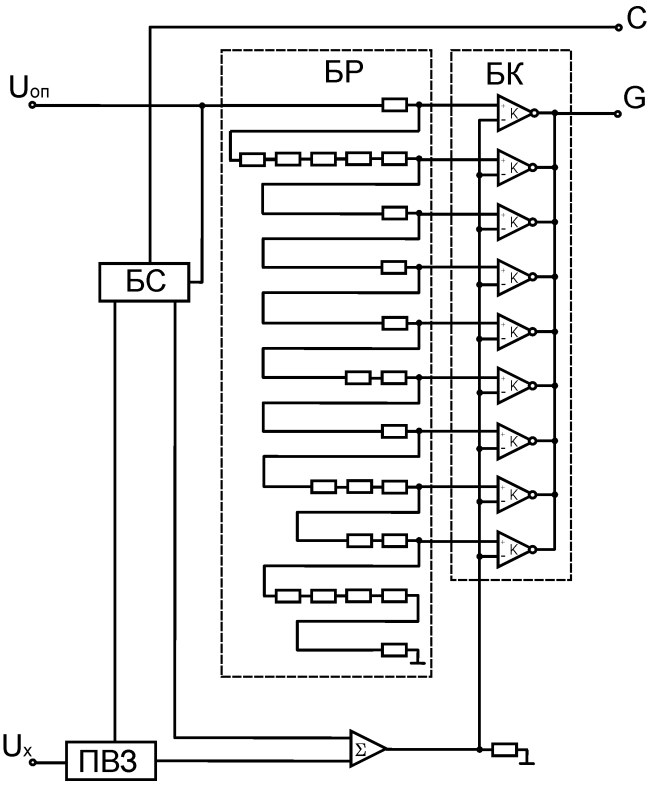


Рис.11.14. Структура оптимізованого АЦП Галуа з пристроєм вибірки та зберігання(ПВЗ).

На рис.11.15 зображена часова діаграма роботи АЦП.

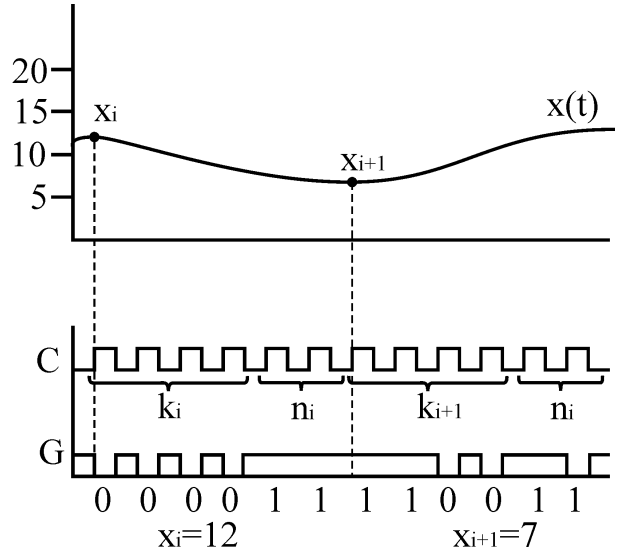


Рис.11.15. Часова діаграма роботи АЦП.

По зростаючому фронту імпульсу синхронізації *C* відбувається запуск перетворення. При цьому з входу U_x зчитується рівень вхідного аналогового сигналу, значення якого запам'ятовується в пристрої вибірки і зберігання, звідки через суматор аналогових сигналів Σ подається на входи імпульсних компараторів. З кожним тактом багатостабільний елемент (БС) формує додаткову скануючу напругу, яка рівна одному кванту АЦП, ця напруга додається до вимірюваного сигналу і викликає спрацювання відповідного компаратора, який формує відповідний нульовий біт коду Галуа. Якщо в точці відповідного кванту відсутній компаратор, то зчитується одиничний біт. Кількість тактових імпульсів необхідних для зчитування коду рівна розрядності АЦП *k* та кількості додаткових бітів захисту *n*.

Код Галуа, який використовується в даному АЦП формується на основі рекурентного рівняння $G_{i+1} = G_i \overline{\oplus} G_{i-n}$ при *n*=4 має наступний вигляд: 1111010110010000. В табл.11.5 наведені відповідності квантів та вихідного коду АЦП без захисту та з захистом від помилок, що дозволяє виявляти помилки передавання інформації та відновлювати цифрові значення.

Таблиця 11.5.

Відповідності значення сигналу та вихідного коду АЦП

Десяткове значення	Відповідність Коду Галуа	Вихідний код з захистом від помилок
0	1111	111101
1	1110	111010
2	1101	110101
3	1010	101011
4	0101	010110
5	1011	101100
6	0110	011001
7	1100	110010
8	1001	100100
9	0010	001000
10	0100	010000
11	1000	100001
12	0000	000011
13	0001	000111
14	0011	001111
15	0111	011110

При побудові АЦП іншої розрядності використовується код Галуа, що формується згідно ключів поданих в табл.11.1.

Кількість компараторів таким чином буде рівною $2^k/2+s$, де s кількість додаткових компараторів, що залежить від кількості біт (табл.11.5) скануючого коду n та значень рекурентного коду Галуа, а тому $s \le n$.

Захист від помилок в даному АЦП базується на рекурентних властивостях коду Галуа. Так, наприклад, для десяткового значення коду рівного 7 (1100) і n=2 (2 додаткових біти) завадозахищений код матиме вигляд 110010 і буде містити в собі 3 кодони Галуа: G_i=(1100); G_{i+1}=(1001); G_{i+2}=(0010), що відповідає десятковим значенням 7,8 і 9. Тому перевірка на наявність помилки в коді зводиться до перевірки відповідності значень рекурентного коду. При розрахунку рекурентних значень формули матимуть вигляд: $G_i = G_{i-1} \overline{\oplus} G_{i-(k+1)}$, $G_{i+1} = G_i \overline{\oplus} G_{i-k}$, $G_{i+2} = G_{i+1} \overline{\oplus} G_{i-k+1}$.

Розрахунок швидкодії даної архітектури АЦП виконується на основі наступної оцінки часу затримки:

$$T_{21}=k+n(t_C+ t_{БС}+t_K)+t_{ПВЗ},$$

де n – число додаткових розрядів коректуючого коду, які формуються в процесі аналого-цифрового перетворення.

Оцінка структурної складності розраховується на основі архітектури (рис.11.14) і прийме наступний вигляд:

$$S_{21}=B_{ПВЗ}+B_{БС}+B_{АНС}+kB_R+kB_K.$$

Проте приведений аналіз АЦП не враховує способу представлення вихідних даних. Так наприклад, представлення вихідних даних послідовним кодом є більш ефективним ніж паралельним, оскільки дозволяє спростити схемотехніку підключення АЦП за рахунок використання однобітної інтерфейсної шини замість *k* розрядної. Тому для більш точної оцінки, доцільно ввести критерій, який дозволяє врахувати кодовий базис, інтерфейсні параметри представлення вихідного коду та наявність захисту даних від помилок. При цьому розрахунок системних характеристик АЦП виконується згідно виразу:

$$\theta = \frac{S_i \prod_{j=1}^{4} F_{i,j}}{T_i / \tau_B}, \qquad (11.1)$$

де $F_{i,j}$ – експертна оцінка характеристик вихідних кодів АЦП згідно табл. 11.6.

Використана мультиплікативна функція (11.1) дозволяє підвищити чутливість врахування таких важливих характеристик АЦП, як спосіб представлення вихідних даних, тип використовуваного ТЧБ, наявність захисту від помилок та можливість одночасного формування миттєвих та інтегральних цифрових значень вхідних сигналів.

Таблиця 11.6.

Експертні оцінки ефективності вихідних кодів АЦП.

№	Пояснення	Ваговий коефіцієнт, F_j
1	Представлення вихідних даних	
	Паралельне	1,0
	Послідовне	1,2
2	Базис	
	Радемахера	1,0
	Унітарний	0,1
	Крестенсона	0,9
	Крейга	1,0
	Галуа	1,0
3	Наявність захисту від помилок	
	Є	1,5
	Нема	1,0
4	Одночасне формування миттєвих та інтегральних значень	
	Є	2,0
	Нема	1,0

Згідно даної оцінки системних характеристик АЦП (11.1) отримаємо наступну гістограму (рис.11.16), яка враховує зміни системних характеристик різних АЦП при зміні розрядності в діапазоні k=8 – 16 біт.

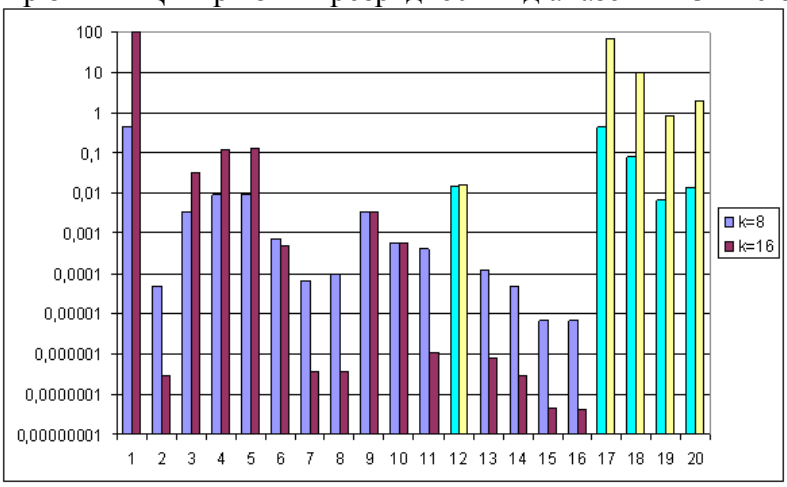


Рис.11.16. Системні характеристики різних класів АЦП.

Рис.11.16 показує, що врахування наявності захисту вихідних кодів від помилок, вихідного ТЧБ та формування вихідних даних у послідовному коді, змінює систему лідерів серед існуючих класів АЦП наступним чином:

1. АЦП паралельного типу в базисі Радемахера;
2. АЦП Галуа з буферним регістром;
3. Скануючий в базисі Галуа;
4. Скануючий в базисі Галуа на імпульсних компараторах з ПВЗ та захистом від помилок;
5. Скануючий в базисі Галуа на імпульсних компараторах;
6. Послідовно-паралельний багатоступінчатий;
7. Конвеєрний;
8. Багататактний послідовно-паралельний;
9. Інтегрально-імпульсний АЦП з імпульсним вихідним кодом в базисі Крейга-Галуа;
10. Сігма-Дельта АЦП.

На основі критерію ефективності схемотехнічного рішення архітектури АЦП (11.1) можна порівняти системні характеристика запропонованої архітектури АЦП Галуа на імпульсних компараторах по відношенню до відомої архітектури №20, шляхом задання значень експертної оцінки переваг: F_1,F_2,F_3, в яких F_1 та F_3 змінюються в діапазоні від 1 до 2. При цьому отримаємо наступну характеристику зміни переваг запропонованої схеми АЦП Галуа (рис.11.17).

На даному графіку при оцінці характеристик використовуються дві шкали одна для АЦП №19 в діапазоні зміни коефіцієнта F_1 від 1 до 2 з

кроком 0,1, а друга для АЦП №20 згідно добутку F_1F_3 в діапазоні зміни добутку коефіцієнтів F_1 та F_3 від 1 до 3 з кроком 0,2, що відповідає таблиці експертних оцінок (табл.11.6), яка враховує наявність захисту вихідних кодів від помилок та представлення вихідних даних у паралельному чи послідовному коді.

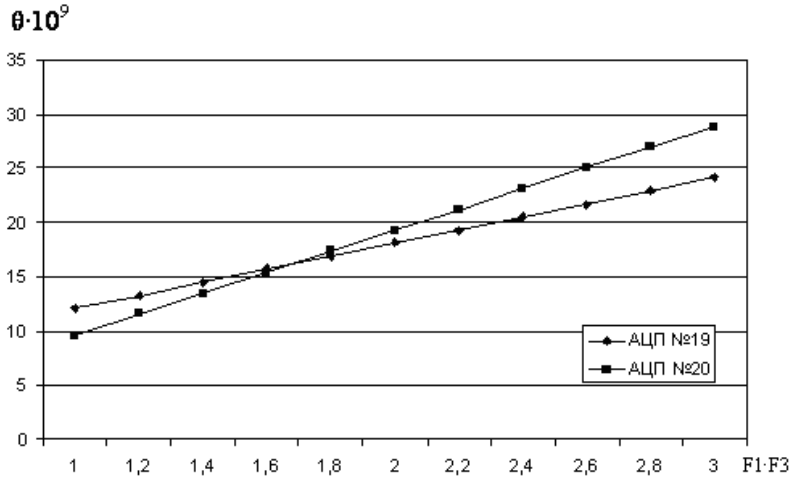


Рис.11.17. Зміна системних характеристик оптимізованого АЦП Галуа по відношенню до відомої схеми (№19, табл.11.4).

На рис.11.18 показана транскрипція даного графіка у вигляді процентного відношення переваг в залежності від зміни експертних коефіцієнтів F_1,F_3.

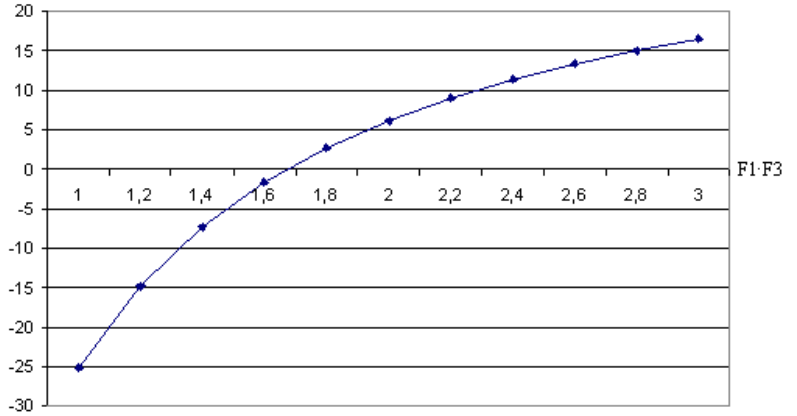


Рис.11.18. Відсоткова оцінка зниження та підвищення рівня системних характеристик АЦП №19 по відношенню до АЦП №20.

З рис.11.18 видно, що при зміні експертних коефіцієнтів переваг F_1 та F_3 у діапазоні 1-1,3 ведення ПВЗ та зчитування $k+n$ розрядів коду Галуа з захистом від помилок погіршує системні характеристики запропонованої схеми АЦП і не компенсується виключенням $2^k/2$ логічних вентилів зі структури АЦП. При експертних оцінках переваг нових функціональних можливостей АЦП №20 в діапазоні 1,3-3 покращення системних характеристик схеми багатоканального АЦП Галуа досягає 20%.

11.4. Кодові шкали Галуа.

11.4.1. Однобітові кодові шкали Галуа.

Кодові шкали Галуа можуть бути однобітові, двохбітові та трьохбітові. При цьому кодові шкали будуються на основі методів виключення гонок при зчитуванні даних шляхом:

- збільшення числа зчитуючи елементів;
- врахування напрямку руху та обертання кодової шкали;
- збільшення числа доріжок шкали.

У першому випадку число зчитуючи елементів збільшується в два рази, при чому здійснюється зменшення половини з них на половину лінійного розміру елемента кодової шкали.

На рис.11.19 приведена структура такого перетворювача лінійних чи кутових переміщень, який містить n – зчитуючих елементів коду Галуа (G_1, G_2 … G_n), n - зчитуючи елементів синхронізації (S_1, S_2 … S_n), блок запису даних P, логічний блок синхронізації запису та дешифратора перетворення Галуа-Радемахера на основі ПЗП.

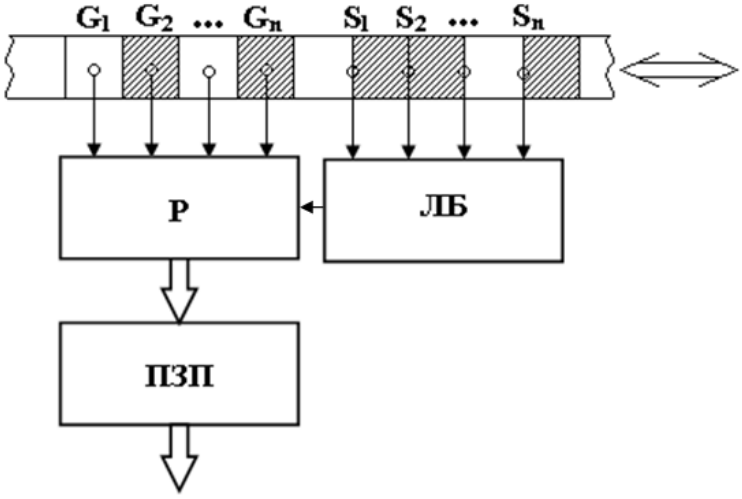


Рис.11.19. Кодова шкала Галуа з подвійним числом зчитуючих елементів.

Логічний блок ЛБ містить формувачі коротких імпульсів, які формуються по фронту наростання або спаду сигналів елементів синхронізації. Причому імпульси запису сигналів інформаційних зчитувачів G_1, G_2 … G_n записуються в Р, коли останні знаходяться у центрі кодових елементів шкали G_i.

Недоліком такого перетворювача э втрата інформації при відключенні живлення. Для відновлення інформації необідно зміщення кодової шкали та її сканування на віддалі не менше половини розміру кодового елемента.

При відомому напрямі руху кодової шкали Галуа або зчитуючих елементів можливе спрощення апаратної реалізації перетворювача шляхом об'єднання нульових та одиничних зчитувачів коду Галуа, та використання одного елементу синхронізації рис.11.20.

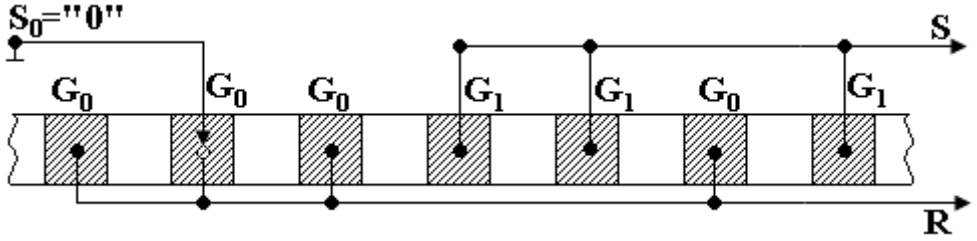


Рис.11.20. Кодова шкала Галуа з одним зчитуючим елементом Z, на який подається потенціал «-».

Дешифратор такої шкали Галуа має структуру, показану на рис.11.21.

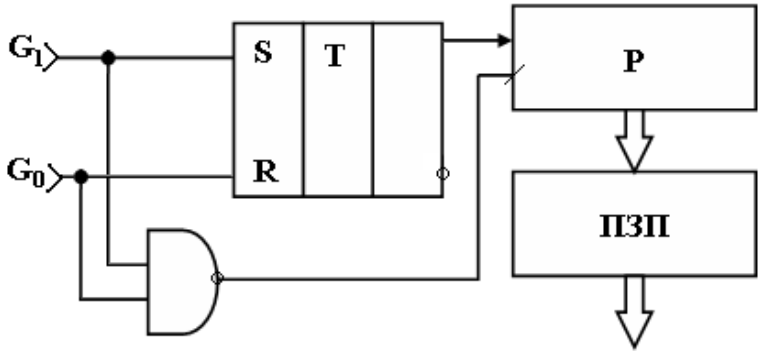


Рис.11.21. Структура дешифратора кодової шкали Галуа з одним зчитуючим елементом S_0.

Описана кодова шкала Галуа характеризується максимальною надійністю оскільки містить тільки один зчитуючий елемент, який виконує функції формування бітів коду Галуа G_0 та G_1 і одночасно виконує функції синхронізатора запису даних в запам'ятовуючий регістр *P*, який реалізований як регістр зсуву на *D* – регістрах. На основі цього принципу

нами запропонований перетворювач переміщення у цифровий код (авт. свідоцтво № 1438545).

Робота перетворювача відбувається наступним чином. У процесі переміщення зчитую чого елемента S_0 по кодовій шкалі у кожні з *SR* – шин у відповідності до КПГ виникають сигнали нульового потенціалу, під дією яких *RS*-трігер перекидається у стан, що відповідає ознаці коду Галуа G_1 або G_0.

При цьому у кожному такті на виході логічного елемента «І-НЕ» виникає сигнал «+», по фронту наростання якого відбувається запис стану RS-трігер у регістр зсуву Р.

Таким чином початкова ідентифікація коду шкали відбувається за n-тактів, а наступні коди Галуа формуються у кожному такті зчитую чого елементу S_0.

Головними функціональними недоліками такого перетворювача є однонаправленість зчитування і можливість виникнення хибних імпульсів зсуву на границях кодових елементів за рахунок дребезга сигналів на шинах SR.

Принципово вплив дребезгу сигналів SR можна ліквідувати застосуванням двох сигнальних зчитувачів зсунутих на віддаль менше половини лінійного розміру кодового елемента. На рис.11.22 показана схема безгоночного електронного модуля формування сигналів управління дешифратором.

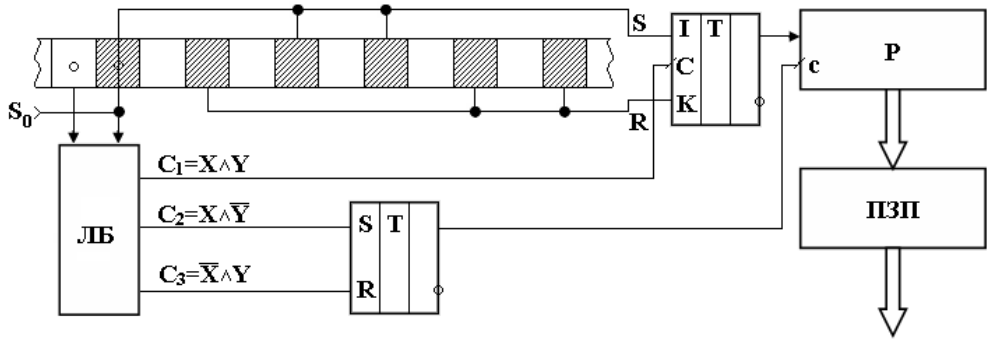


Рис.11.22. Схема безгоночного зчитування даних кодової шкали Галуа.

11.4.2. Двохбітова шкала Галуа.

Двох бітові кодові шкали Галуа характеризуються значними спрощеннями системи зчитування. Структура такої шкали Галуа з n-інформаційних (G_1, G_2 ... G_n) та одного синхронізуючого елемента S_0 показана на рис.11.23.

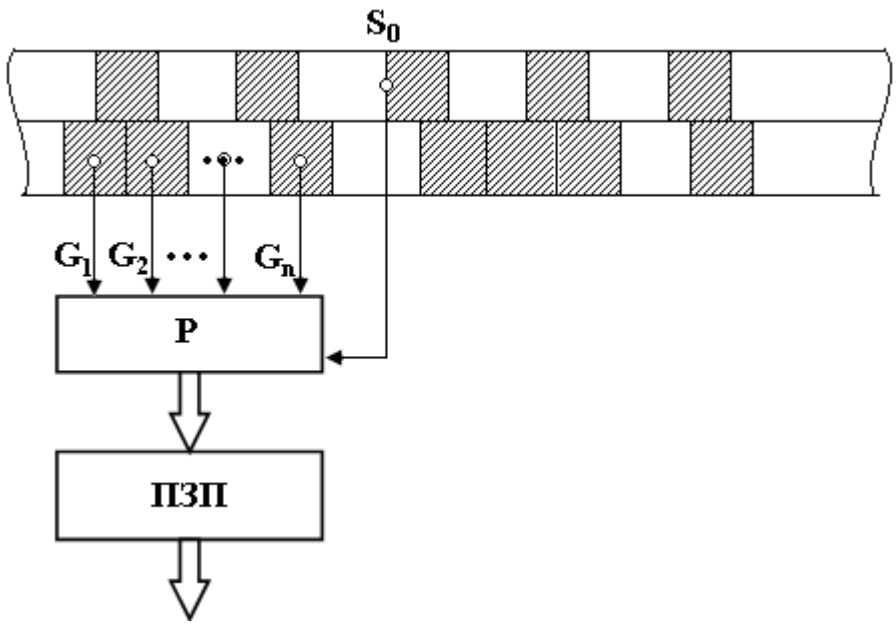


Рис.11.23. Двохбітова кодова шкала з паралельним зчитуванням коду Галуа.

11.4.3. Квазітрійкова шкала Галуа.

Трьохбітові квазітрійкові кодові шкали Галуа створюються на основі квазістаціонарного КПГ з захисними інтервалами (рис.11.24).

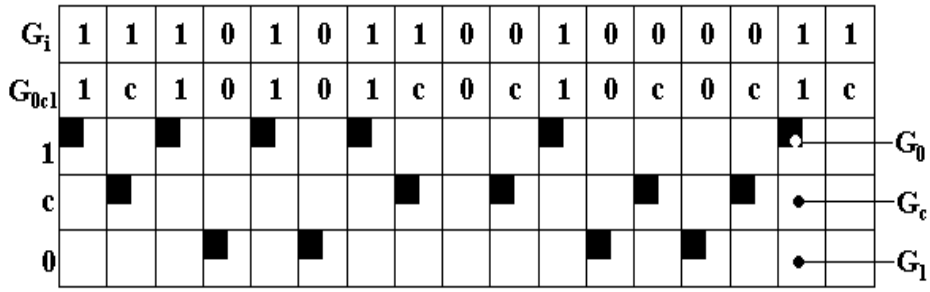


Рис.11.24. Трохбітова кодова шкала Галуа G_i – інформаційний код; G_0 – квазітрійковий КПГ; (1СО) – кодова шкала.

На рис.11.25 показана бездребезгова схема формування імпульсів синхронізації запису даних.

Позитивною характеристикою таких кодових шкал є нечутливість до дребезгу сигналів зчитування самосинхронізація та незалежність числа

кодових доріжок від діапазону зчитування. Недоліком є однонаправленість зчитування даних.

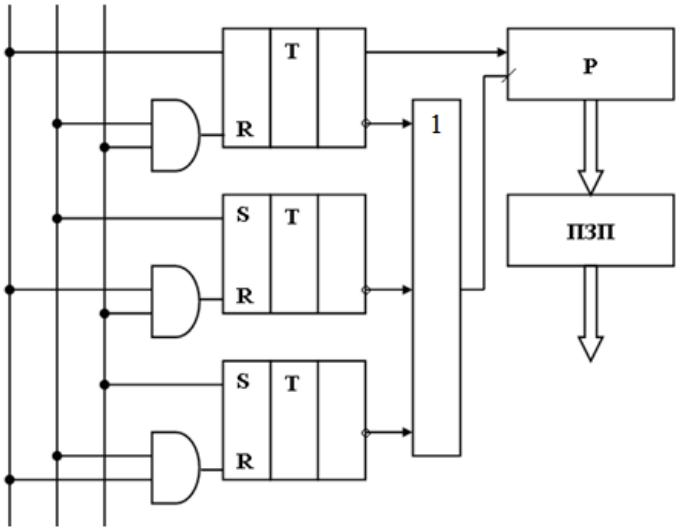


Рис.11.25. Бездребезговий дешифратор трьохбітової шкали Галуа.

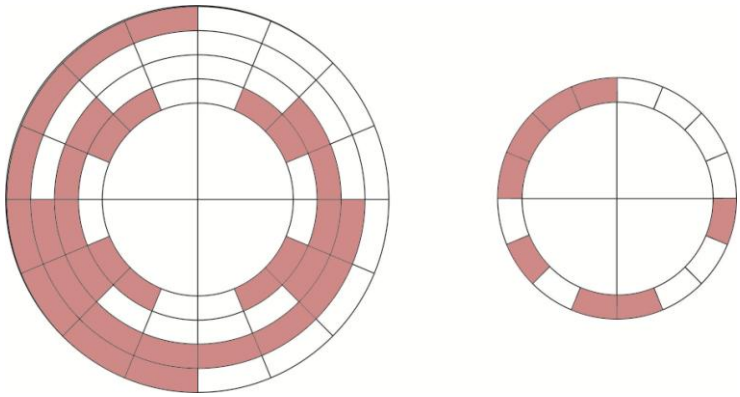
Описана трьохбітова шкала Галуа впроваджена і успішно експлуатується на дзвоновій дискретно-динамічній установці для точного відтворення об'ємів і витрат газу, яка є державним первинним еталоном одиниці об'єму та об'ємної витрати газу ДЕТУ 01-03-96.

11.5. Кодові диски у базисі Галуа.

Відомі приклади використання кодових шкал на основі псевдовипадкових послідовностей Галуа для побудови перетворювачів кута повороту у цифровий код. Головною перевагою таких перетворювачів у порівнянні з кодовими дисками Грея є можливість підвищення точності перетворювача при однаковій точності зчитування кодового елемента і діаметрі диска або відповідного зменшення діаметра диска при збереженні точності перетворення.

Головним обмеженням таких кодових шкал у полярних координатах є виникнення хибних показів на передніх значеннях кодових елементів, які відсутні в шкалах на основі кодів Грея. Що потребує вбудову додаткової доріжки і одного сенсора для синхронізації зчитування кодових значень кута повороту.

Для порівняння вказаних-типів кодових шкал опишемо їх параметри аналітично на основі рис.11.26 , де а) відповідає чотирьохбітовому диску Грея, а б) – одно бітовому диску Галуа (11110110010000).



а) б)

Рис.11.26. Кодові шкали Грея та Галуа.
- значення одиниці, - значення нуля

Отже діапазон кодування обох перетворювачів співпадає і рівний: $N = 2^n;\ \ n = 4;\ \ N = 16$, де n – число доріжок кодової шкали Грея, яка відповідає числу зчитуваних доріжок (кодону) $n_1=n_2$ однаковому для обох перетворювачів.

Число кодових елементів перетворювача Грея та Галуа відповідно дорівнює

$$M_1 = \frac{n \cdot N_1}{2}; \qquad M_2 = \frac{N_1}{2}.$$

Діаметри кодових дисків при $N_1 = N_2$ і однаковій точності зчитування відповідно рівні

$$d_1 = d_{\min} + n\Delta d; \qquad d_2 = d_{\min} + \Delta d,$$

де Δd - ширина доріжки кодового диска.

Мінімальний кут для розміщення кодового елемента повинна задовольняти умові

$$\Delta\varphi_{\min} = \frac{\pi d_{\min}}{2^n}.$$

Мінімальна довжина дуги кодового елемента повинна задовольняти умові

$$\Delta L \geq \Delta d = R_0 = 1 \qquad (11.2)$$

для деякого уніфікованого диска з радіусом $R_0 = 1$ (рис.11.26), при умові, що форма кодового елемента близька до квадратної.

Тоді умова виграшу в діапазоні квантування кутових величин для диска Галуа по відношенню до диска Грея можна визначити з умов

$$N_1 = 2\pi K_1 R_0; \qquad (11.3)$$

$$N_1' = 2\pi(K_1 R_0 + nR_0), \qquad (11.4)$$

де $K_1R_0 = d_1$ - діаметр першої внутрішньої доріжки диска; N_1' - діапазон квантування диска Галуа, реалізованого на n-тій зовнішній доріжці диска Грея.

Підставляючи (11.2) і (11.3) в (11.4) отримаємо вираз

$$N_1' = \breve{E}[N_1 + 2\pi n],$$

на основі якого побудована таблиця 11.7, з якої видно, що виграш у збільшенні діапазону квантування кутових величин Δn спостерігається при $n \le 0$.

Таблиця 11.7.

Розрахункові дані кодових дисків Галуа

n	N_i	N_i/N_1
2	16	4.14
3	26	3.35
4	41	2.57
5	63	1.98
6	101	1.86
7	171	1.34
8	306	1.19
10	433	1.09

Наприклад порівняємо габаритні параметри 10-бітних кодових дисків Грея та Галуа. Нехай $n = 8; \quad \Delta L = 2mm; \quad N_1 = 256$ тоді

$$\rho_0{}^n = \frac{256}{2\pi} = 81mm; \quad N_1 = 81 + 16 = 97mm;$$

$N_i / N_1 = 1.19,$ а число кодових елементів замість $N_1 = 4 \cdot 256 = 1024$ в дисках Грея буде рівне 128 у дисках Галуа.

11.6. Формувачі широтно-модульованих сигналів на основі рекурентних кодів Галуа.

До класу формувачів широтно-модульованих сигналів відносяться перетворювачі аналог-код розгортую чого типу, які містять проміжні представлення аналогових величин та кутів повороту у вигляді широтно-імпульсної модуляції (ШІМ) та імпульсно-частотної модуляції (ІЧМ) сигналів. Безпосереднє передавання таких сигналів в умовах впливу інтенсивних завад неефективне у зв'язку з появою дребезгу фронтів наростання та спаду в ШІМ-сигналах. Аналогічно відбувається розмивання та спотворення форми ІЧМ-сигналів.

На рис.11.27 показані часові діаграми затухання та спотворення форми ШІМ та ІЧМ-сигналів. Таким чином, в умовах, коли потужність завад

у лініях зв’язку, по яких передаються названі сигнали, наближається або перевищує потужність сигналів, їх виявлення та виділення стає практично неможливим. Тим більше, недоступним є цифрове відтворення цих сигналів з необхідною точністю параметрів їх тривалості або частоти слідування.

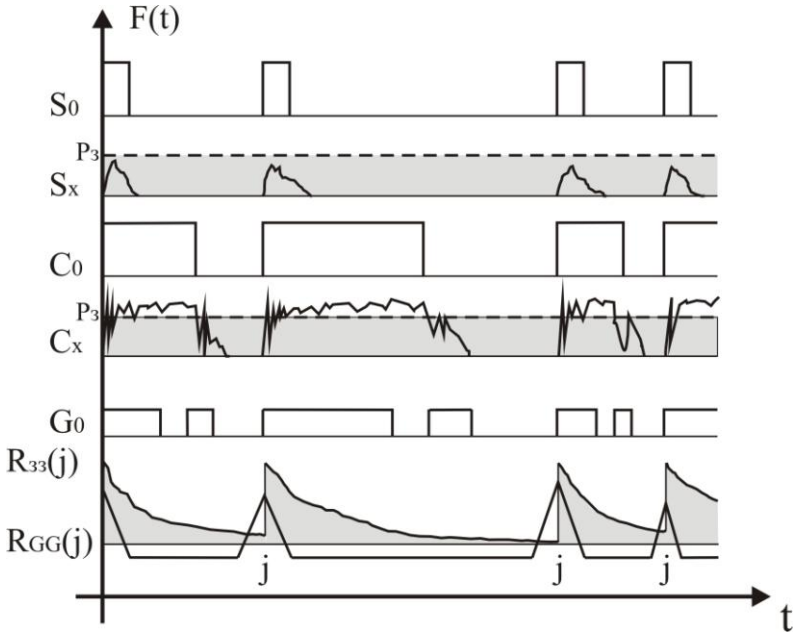


Рис.11.25. Часові діаграми умов передавання ІЧМ(S), ШІМ(C) та ШПС(G) сигналів.

Представлення такого класу сигналів у вигляді широтно-модульованих бінарних кодів поля Галуа дозволяє на 1-2 порядки підвищити дальність їх передавання та забезпечити надійне виділення на основі особливих кореляційних властивостей рекурентних послідовностей кодів поля Галуа.

Теоретичної основою ефективного застосування широтно-модульованих сигналів базису Галуа є виконання фундаментальних обмежень К. Шеннона, які для різних способів модуляції сигналів теоретично обмежують віддаль їх надійного передавання по лініях зв’язку з різним ступенем затухання сигналів в умовах впливу інтенсивних завад (рис.11.28).

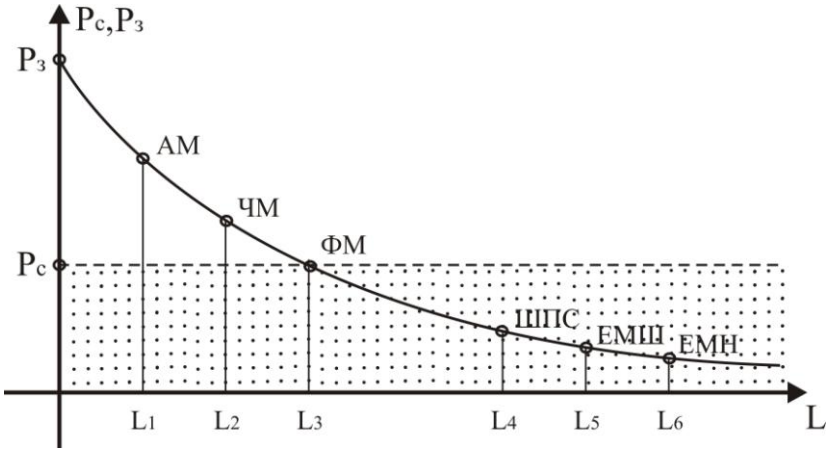


Рис.11.28. Умови передавання модульованих сигналів при фундаментальних обмеженнях Шеннона.

На рис.11.28 введені наступні позначення:

$P_с$, $P_з$ – відповідно потужність сигналів та завад;

$L_1 \div L_6$ - максимальна віддаль передавання та надійного виділення сигналів;

АМ, ЧМ, ФМ, ШПС, ЕМШ, ЕМН – відповідно: амплітудна, частотна, фазова, широкосмугова та ентропійна модуляція сигналів (ЕМШ, ЕМН згідно оцінки ймовірнісної ентропії Шеннона та оцінки кореляційної ентропії Николайчука).

Умови фундаментальних обмежень К.Шеннона для різних способів модуляції або маніпуляції сигналів описується наступними виразами:

1. $AM(L_1) \Rightarrow \dfrac{P_c(f_{\max})}{P_ç(f_{\max})} \geq 2$;

2. $\times M(L_2) \Rightarrow \dfrac{P_c(\Delta f)}{P_ç(\Delta f)} \geq 2$;

3. $\hat{O}M(L_3) \Rightarrow \dfrac{P_c(f_i)}{P_ç(f_i)} \geq 2$;

4. $\text{ØÏÑ}\quad (L_4) \Rightarrow \dfrac{R_{CC}(j)}{P_{ÇÇ}(j)} \geq 2$

5. $EM(L_5) \Rightarrow \dfrac{I_{ÑØ}}{I_{ÑØ}} \geq 2$;

6. $EM(L_6) \Rightarrow \dfrac{I_{Ñí}}{I_{Ñí}} \geq 2$,

де $f_{\max}$ - максимальна смуга частот ліній зв'язку, Δf - ширина смуги девіації частоти, f_i - несуча частота фазоманіпульованих сигналів, $R_{CC}(j), P_{ÇÇ}(j)$ - відповідно значення автокореляційних функцій сигналу та завади на інтервалі зсуву j, $I_{c,} I_ç$ - відповідно кореляційна міра ентропії сигналу та завади.

Можливість виділення та приймання ШПС, які належать до класу рекурентних послідовностей кодів Галуа, базується на тому факторі, що

автокореляційна функція завади $R_{ÇÇ}(j)$ затухає, наближаючись до експоненти, а автокореляційна функція ШПС $R_{CC}(j)$ в точці часового зсуву j має особливий пік, який перевищує значення $R_{ÇÇ}(j)$ при аналогічному зсуві (рис.11.29).

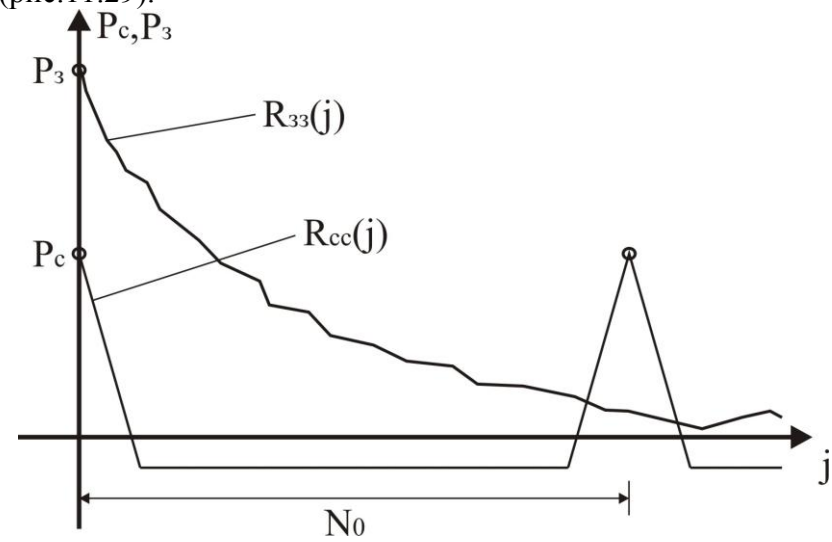


Рис.11.29. характеристики затухання автокореляційних функцій ШПС та завади.

На рис.11.30 показана структура перетворювача, який формує вихідні модульовані за періодом сигнали коду поля Галуа. Перетворювач містить: вхідну шину 1, з'єднану із входом дільника частоти 2 і першим входом логічного елемента «І» 3, другий вхід якого з'єднаний з виходом генератора імпульсів 4 і лічильним входом дільника частоти 5 зі змінним коефіцієнтом ділення, а вихід – з'єднаний з входом лічильника імпульсів 6, вхід установки в «0» якого з'єднаний з виходом дільника частоти 2 і входом запису керованого дільника 5, а виходи – з інформаційними входами дільника частоти 5, вихід якого через генератор рекурентного коду Галуа 7 з'єднаний з вихідною шиною 8.

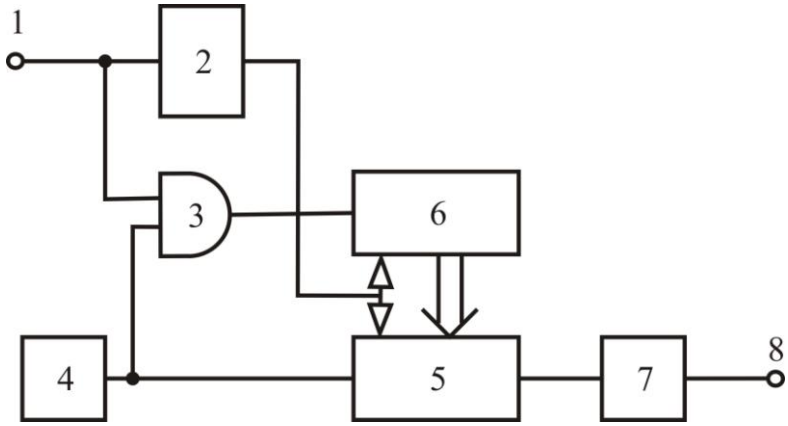


Рис.11.30. Структура формувача модульованого за періодом сигналів коду поля Галуа.

Модуляція періоду сигналів шумоподібного коду Галуа здійснюється шляхом зміни частоти імпульсів, які поступають на вхід генератора коду Галуа 7, реалізованого, як це показано у розділі 8, на основі регістру зсуву зі зворотнім логічним зв'язком за модулем *P*.

Алгоритм роботи даного перетворювача описується наступними аналітичними функціями.

Після приходу *m* ШІМ-сигналів на шину 1 у лічильнику 6 буде записаний код K_j. код суми значень їх тривалостей:

$$K_j = F_0 \cdot \sum_{i=1}^{m} t_i \,,$$

де F_0 - частота імпульсів генератора 4; t_i - тривалість *i*-го ШІМ-сигналу.

У дільник зі змінним коефіцієнтом ділення 5 переписується значення K_j / m шляхом відкидання *m* молодших розрядів суми K_j і здійснюється скид лічильника 6 у нульовий стан.

На виході дільника 5 формується імпульси з частотою

$$F_j = \frac{m \cdot F_0}{K_j} = m^{-1},$$

які тактують роботу генератора коду Галуа типу М-послідовності з періодом

$$T_j = \frac{N_0}{F_j} = \frac{N_0}{m} \sum_{i=1}^{m} t_i \qquad (11.5)$$

де N_0 - довжина М-послідовності.

Частота F_0 вибирається в залежності від допустимої похибки Δt_d вимірювання тривалості ШІМ-сигналу за формулою

$$F_0 \geq \frac{t_{\max} - t_{\min}}{\Delta t_d} \cdot F_1 \cdot h_{\min},$$

де $h_{\min}$, F_1 - відповідно мінімальна скважиність частота формування вхідних ШІМ-сигналів.

Вибір m визначається частотним спектром вхідних сигналів, а також швидкодією АЦП, який формує ШІМ-сигнал

$$m \leq F_{\max} \cdot t_{ïð},$$

де $F_{\max}$ - максимальна частота у спектрі вхідного сигналу; $t_{ïð}$ - час, що витрачається на широтно-імпульсне формування сигналу.

Довжина М-послідовності рекурентного коду поля Галуа вибирається, виходячи з вимог необхідної завадозахищеності і ширини смуги пропускання частот ліній зв'язку.

При визначенні величин m і N_0 необхідно забезпечувати формування заданого числа P_0 періодів М-послідовності з безперервною часовою розгорткою, достатньою для кореляційного виділення модульованих М-сигналів на всьому діапазоні зміни їх періоду.

$$P_0 \geq \frac{T_n}{T_{j\max}}, \qquad (11.6)$$

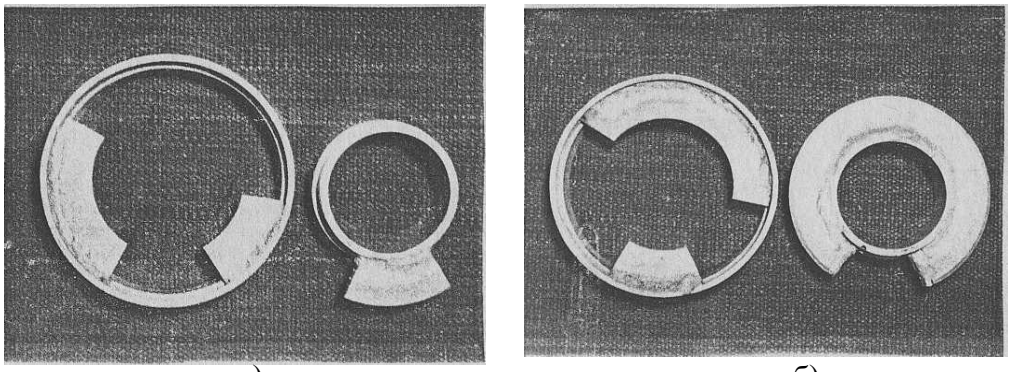
де $T_n = m/F_1$ - інтервал між зміною частоти імпульсів, які тактують роботу генератора коду поля Галуа.

Підставляючи значення T_n і (11.5) у (11.6) отримаємо умову вибору P_0 у вигляді

$$P_0 \geq \frac{m}{N_0 \cdot F_1 \cdot t_{\max}}.$$

Описаний принцип формування модульованих за періодом сигналів коду Галуа застосований при створенні систем контролю швидкості обертання турбобура по гідравлічному каналу з глибини 6-9 тис.м при бурінні зверх глибокої свердловини на Кольському півострові з проектною глибиною 12 тис. м.

Системи на основі переривання потоку промивної рідини частотно-модульованими імпульсами не дозволяли виділити такі сигнали на фоні гідравлічних завад, що створювалися потужними наземними насосами, з глибин більше 2-3 тис. м. Тому автором сумісно з С.М. Іщеряковим створений гідравлічний генератор модульованих М-сигналів коду поля Галуа у вигляді вставки у турбобур, який показано на рис.11.31.



а) б)

Рис.11.31. Гідравлічні генератори широтно-модульованих М-сигналів коду поля Галуа (1110100). а) – з виступом на роторі, б) – з впадиною на роторі.

Проста конструкція описаного формувача сигналів забезпечила його надійну роботу на забої при бурінні глибоких свердловин та забезпечила досягнення високого рівня завадозахищеності у гідравлічних каналах з інтенсивним рівнем шумів.

Виявлення та виділення описаного класу сигналів, що відповідають модульованим кодам поля Галуа здійснюється за допомогою кореляційних спецпроцесорів.

РОЗДІЛ 12

ЗМЕНШЕННЯ НАДЛИШКОВОСТІ ДАНИХ У БАЗИСІ ГАЛУА

12.1. Стиснення даних на основі логіко-статистичних інформаційних моделей та кодів Галуа.

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 нестаціонарні 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.

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 $\overline{G}$ (де n_g – розрядність кодону Галуа).

Перша ЛСІМ фіксує вихід за границі апертури "норми" амплітудних значень характеристичних параметрів P_{ijk} (рис.12.1) згідно аналітичного виразу:

$$g = \begin{cases} G, & P_{ijk} > E_5 \\ \overline{G}, & P_{ijk} \le E_5 \end{cases},$$

де E_1 – границя апертури амплітудних значень;

G_i - біт коду Галуа.

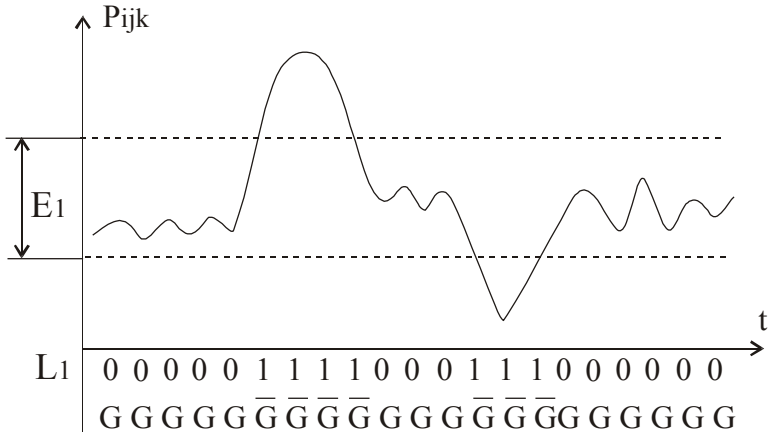


Рис.12.1. Перша ЛСІМ.

Друга ЛСІМ фіксує зміну динаміки характеристичного параметру P_{ijk}, амплітуда якого не виходить за границі апертури E_1 (рис.12.2) згідно наступного аналітичного виразу:

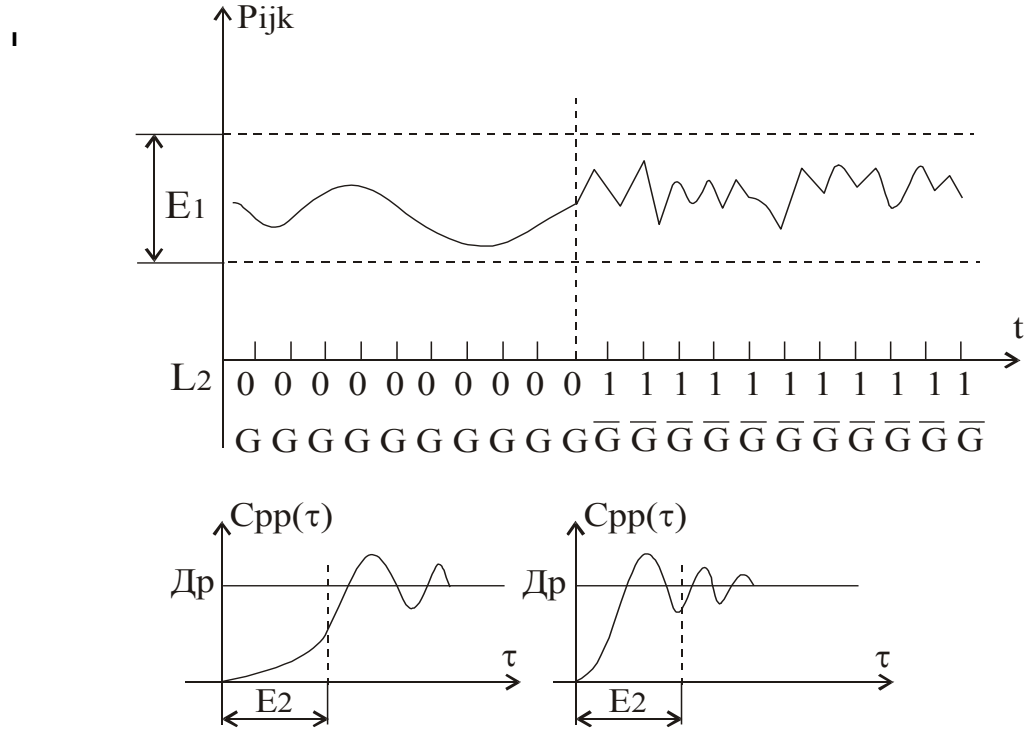


Рис.12.2. Друга ЛСІМ.

$$L_2 = \begin{cases} G, & E < \varepsilon_0 \\ \overline{G}, & E \geq \varepsilon_0 \end{cases},$$

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

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

Третя ЛСІМ фіксує зміну знаку коефіцієнта взаємокореляції між двома характеристичними параметрами P_{ijk} і q_{ijk} з амплітудами, що не виходять за межі апертури (рис.12.3), один з яких q_{ijk} вважається контрольним або еталонним згідно виразу:

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

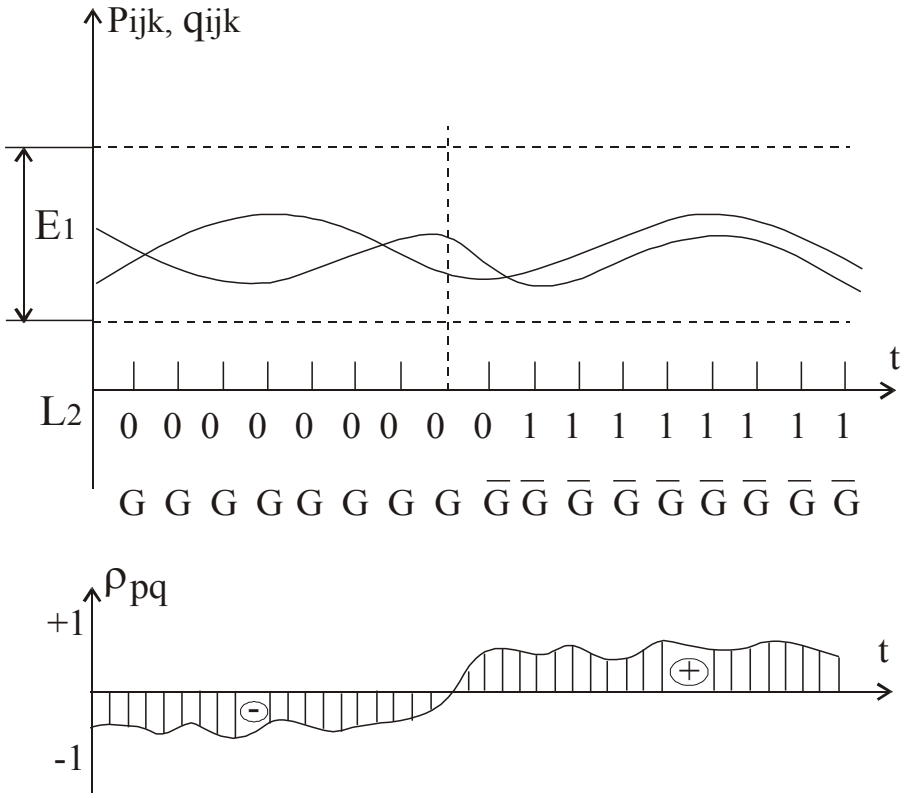


Рис.12.3. Третя ЛСІМ.

Четверта ЛСІМ базується на спектральному аналізі контрольованого сигналу. Множина E_1 визначає спектральний склад сигналу, який відповідає нормальному режиму роботи ДІ (рис.12.4):

$$L_4 = \begin{cases} G, & E_1 = w, \\ \overline{G}, & E_1 \neq w. \end{cases}$$

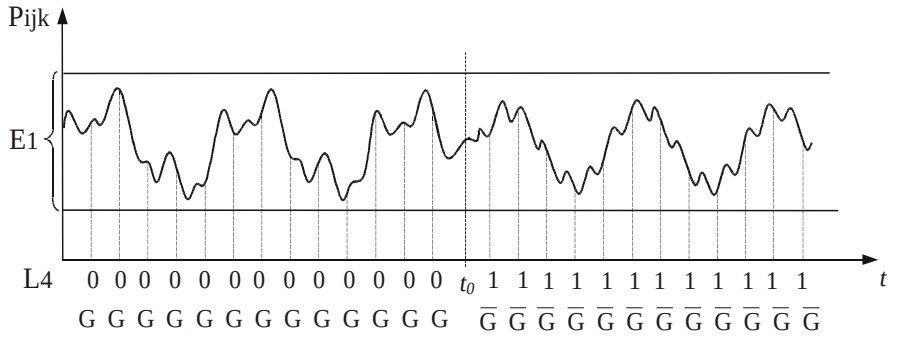


Рис.12.4. Четверта ЛСІМ.

П’ята ЛСІМ базується на матриці коефіцієнтів взаємокореляції 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.12.5), яка оцінює загальний стан багатоканального ДІ:

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

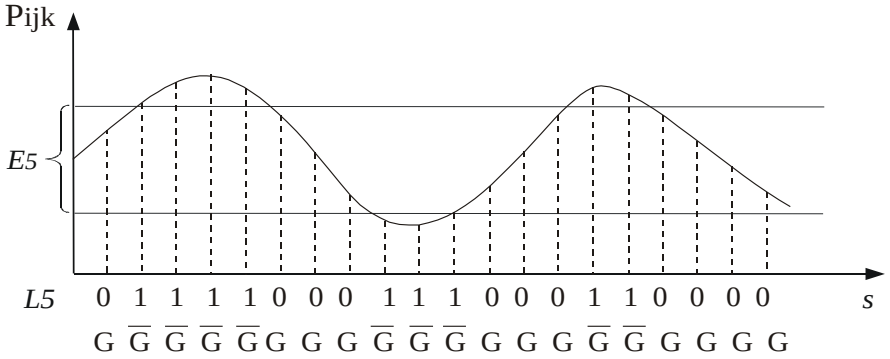


Рис. 12.5. П’ята ЛСІМ на основі глобальної дисперсії.

Шоста ЛСІМ базується на оцінці відхилення значення контрольованого параметру від еталонного. Задається ε_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 відхилення значень параметра від еталона реєструється інвертований біт Галуа.

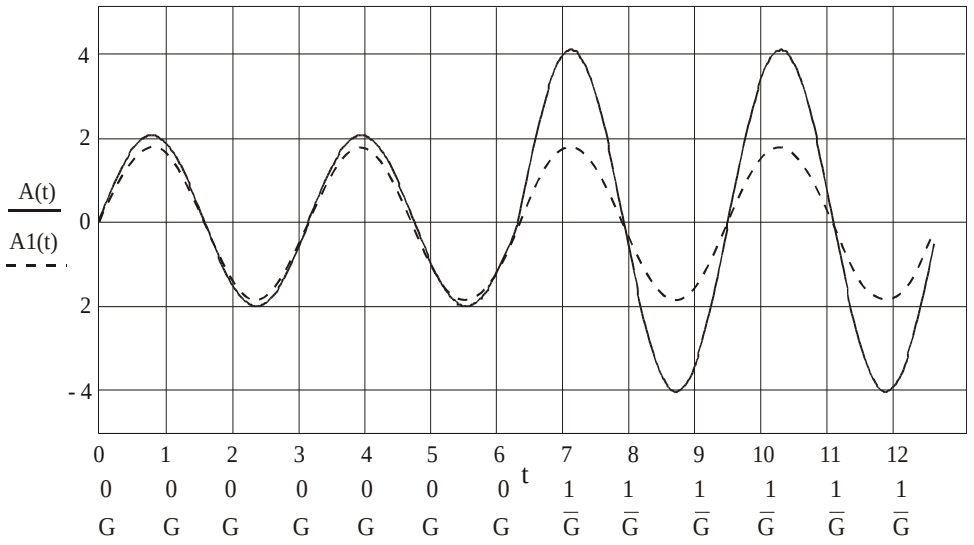


Рис.12.6. Шоста ЛСІМ.

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, Хеммінгових та інших інформаційних моделей ДІ.

Важливу інформацію про стан екологічної безпеки несуть інтегральні моделі семантичних та інформаційних станів ДІ. 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 Δt_0 - розрахунковий інтервал часової дискретизації.

12.2. Теорія та застосування базисних функцій кодів поля 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.

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

В основу принципу дельта-модуляції покладена процедура кодування на основі вибору такого кроку квантування сигналів по рівню δ та кроку дискретизації Δt, при якому виконується умова:

$$\Delta_i - \begin{cases} 1, & x_i - x_{i-1} = +1 \\ 0, & x_i - x_{i-1} = -1 \\ 0,1\ldots & x_i - x_{i-1} = 0 \end{cases}, \qquad (12.1)$$

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 (12.1).

Неповнота системи базисних функцій методу дельта-модуляції обумовлює головний недолік названого методу, який полягає в тому, що при невиконанні умови (12.1) 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. 12.7.

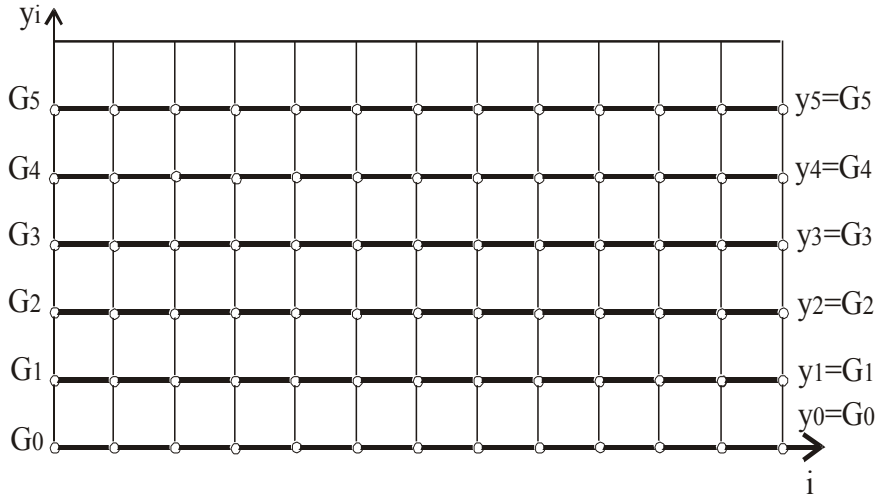


Рис. 12.7. Елементарні базисні лінійні функції Галуа.

З рис 12.7 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 С.

Для того, щоб уникнути даного недоліку, кожна з функцій кодується послідовністю Галуа, яка генерується різним ключем: для $n_g = 4$ (табл.12.1).

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

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

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

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.

Таблиця 12.1.

Фазові базисні функції.

Назва	Кодон	Послідовність Галуа
G_0	0000	0 0 0 0 1 0 1 0 0 1 1 0 1 1 1 1
G_1	1000	1 0 0 0 0 1 0 1 0 0 1 1 0 1 1 1
G_2	1100	1 1 0 0 0 0 1 0 1 0 0 1 1 0 1 1
G_3	1110	1 1 1 0 0 0 0 1 0 1 0 0 1 1 0 1
G_4	1111	1 1 1 1 0 0 0 0 1 0 1 0 0 1 1 0
G_5	0111	0 1 1 1 1 0 0 0 0 1 0 1 0 0 1 1
G_6	1011	1 0 1 1 1 1 0 0 0 0 1 0 1 0 0 1
G_7	1101	1 1 0 1 1 1 1 0 0 0 0 1 0 1 0 0
G_8	0110	0 1 1 0 1 1 1 1 0 0 0 0 1 0 1 0
G_9	0011	0 0 1 1 0 1 1 1 1 0 0 0 0 1 0 1
G_{10}	1001	1 0 0 1 1 0 1 1 1 1 0 0 0 0 1 0
G_{11}	0100	0 1 0 0 1 1 0 1 1 1 1 0 0 0 0 1
G_{12}	1010	1 0 1 0 0 1 1 0 1 1 1 1 0 0 0 0
G_{13}	0101	0 1 0 1 0 0 1 1 0 1 1 1 1 0 0 0
G_{14}	0010	0 0 1 0 1 0 0 1 1 0 1 1 1 1 0 0
G_{15}	0001	0 0 0 1 0 1 0 0 1 1 0 1 1 1 1 0

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.12.8.

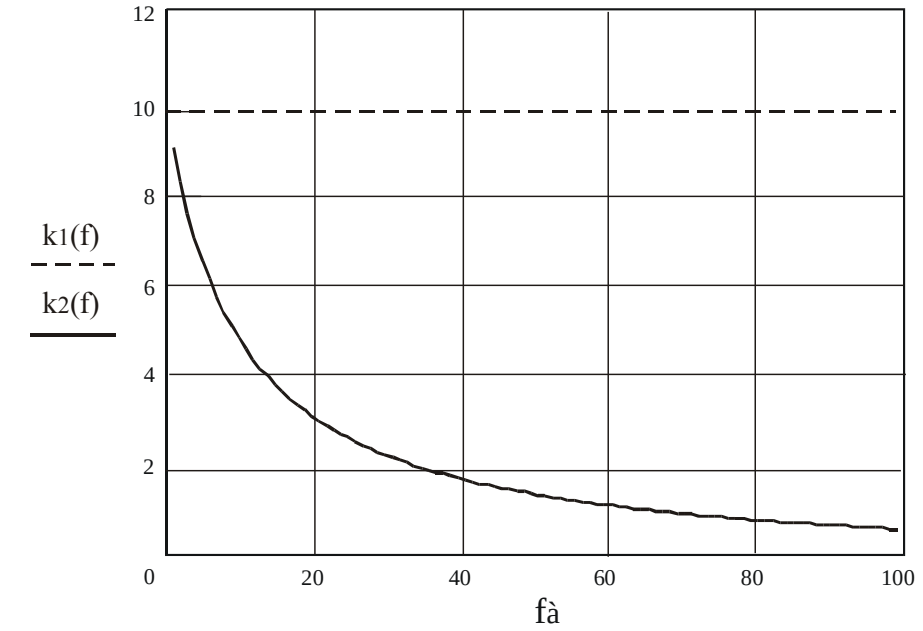


Рис.12.8. Залежність коефіцієнта стиснення від кількості активних відліків.

Практичним застосуванням запропонованого методу є кодування станів об’єктів керування.

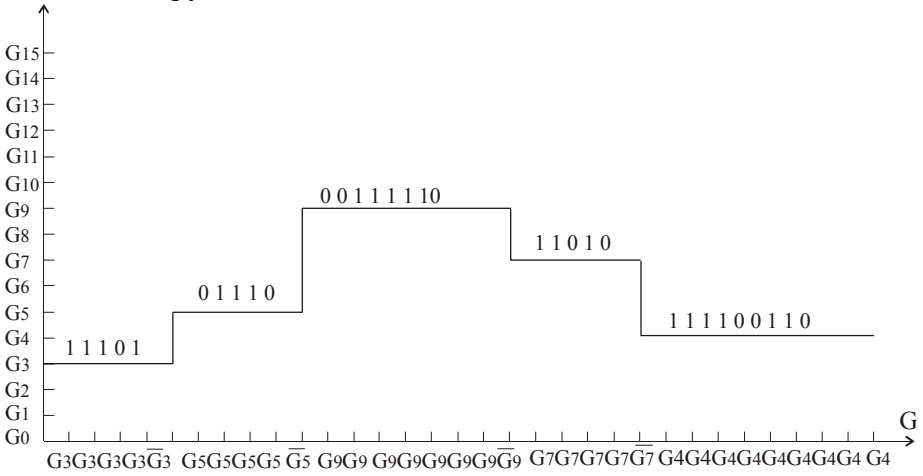


Рис.12.9. Приклад кодування на основі базисних функцій Галуа нульового порядку.

Базисні функції Галуа нульового порядку доцільно використовувати для кодування станів об’єктів керування. Розглянемо приклади: 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.12.10).

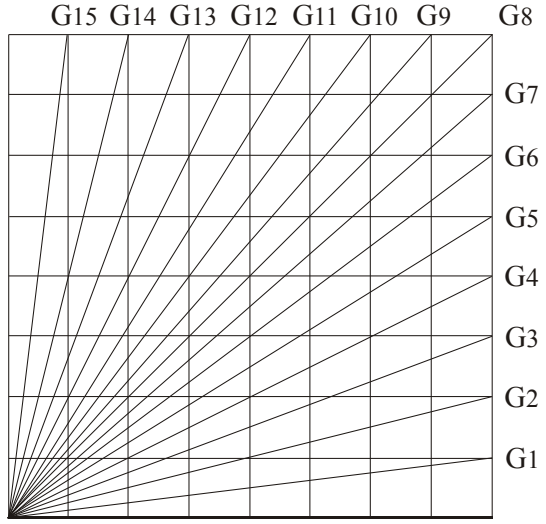


Рис.12.10. Базисні функції Галуа першого порядку.

При цьому біт-орієнтоване кодування дискретних функцій в базисі Галуа однозначно можливе для стрибкоподібних і лінійно-наростаючих функцій, що ілюструється прикладом (рис.12.11).

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.

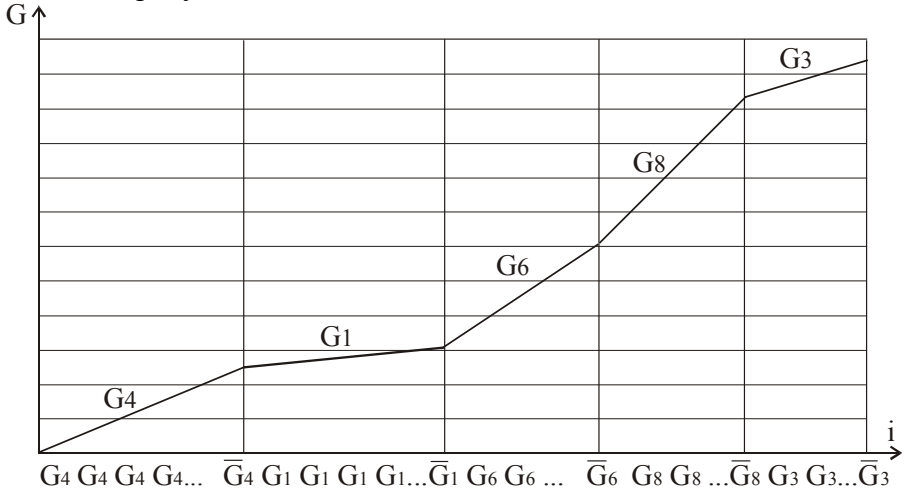


Рис.12.11. Кодування на основі базисних функцій першого порядку.

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.

12.3. Стиснення даних у базисі Крестенсона-Галуа.

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:

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

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

Діофантове рівняння:

$$x_i \equiv b_i (\text{mod } p),$$

або операція прямого кодування по залишках:

$$b_i = res\ x_i (\text{mod } p),$$

де res – символ операції отримання залишку.

Зворотня операція:

$$x_i = \overset{\vee}{E}\left[\frac{x_{i-1} - b_i}{p} + 0{,}5\right] \cdot p + b_i, \qquad (12.2)$$

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

Умова однозначності кодування методом залишків виконується, якщо

$$\Delta x_{\max} \le \frac{p-1}{2}, \qquad (12.3)$$

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

Інформація на рис.12.12 закодована у вигляді залишків. При виконанні умови (12.3) процес x_i можна однозначно представити послідовністю залишків b_i.

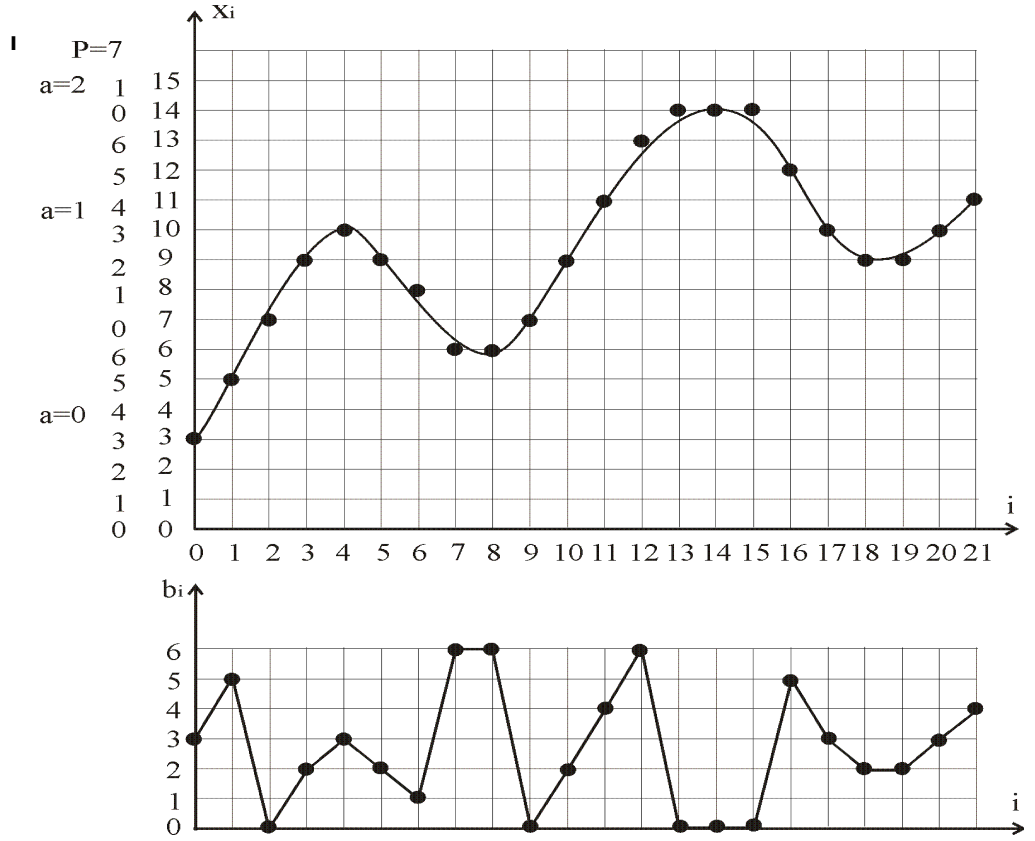


Рис.12.12. Кодування аналогового сигналу методом залишків.

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

Розрядність коду x_i визначається за формулою Хартлі:

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

де n – розрядність двійкового коду для представлення величини x_i;

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

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

Розрядність коду залишків b_i визначаємо за формулою:

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

Отже, формула для коефіцієнту стиснення буде мати наступний вигляд:

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

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:

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

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

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

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

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

$$x_4 = \overset{\vee}{E}\left[\frac{9-3}{7} + 0.5\right] \cdot 7 + 3 = 10;$$ 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.12.12 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.12.13.

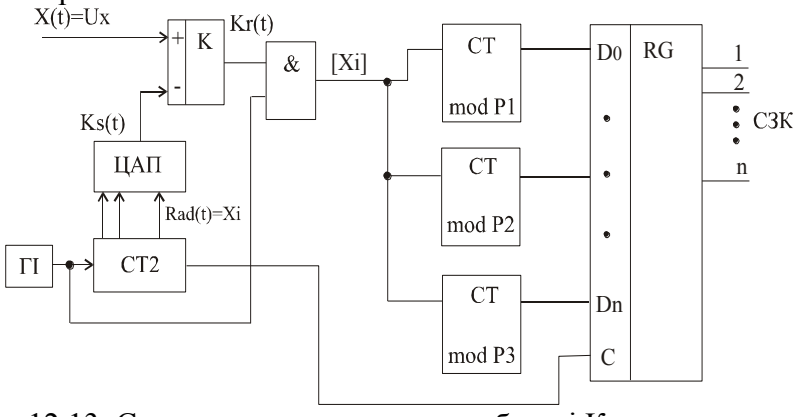


Рис.12.13. Структурна схема кодера в базисі Крестенсона по модулях *P1, P2, P3*.

Робота представленої структурної схеми (рис.12.13) відбувається наступним чином: в початковий момент, який відповідає часу $t=0$, лічильник СТ2 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:

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

Високочастотний сигнал, який модулює генератор ГІ, подається одночасно на вхід лічильника СТ2 і на один з входів логічної схеми "І". При цьому в процесі формування двійкового коду на виходах лічильника СТ2, 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 $x(t) \le Ks(t)$.

В момент виконання умови $U_x \le Ks(t)$ на виході компаратора, який підключений до входу логічного елемента "І", формується нульовий сигнал і, фактично, формується широтно-імпульсна функція базису Крейга $Kr(t)$.

В цей момент на виході логічного елемента "І" припиняється формування числа імпульсів унітарного коду $[x_i] = x(t)$, що представляє собою код унітарного базису x_i.

Для перетворення унітарного коду x_i в систему залишкових класів, унітарний код x_i поступає на лічильники, які працюють по модулю *P1, P2, P3*. На паралельних виходах лічильників одержуємо код в системі залишкових класів по вибраних модулях. Запис даних в регістр пам'яті здійснюється сигналом переносу лічильника СТ2.

Перевагою використання аналого-цифрового перетворення (АЦП) в СЗК є незалежне утворення інформаційних розрядів, що дає можливість їх паралельної обробки.

Перетворення аналогового сигналу методом залишків по одному модулю (рис.12.14) відбувається аналогічно перетворенню по трьох модулях (рис.12.13).

Коефіцієнт стиснення методом залишків залежить від вибраного модуля P та розрядності даних (рис.12.15).

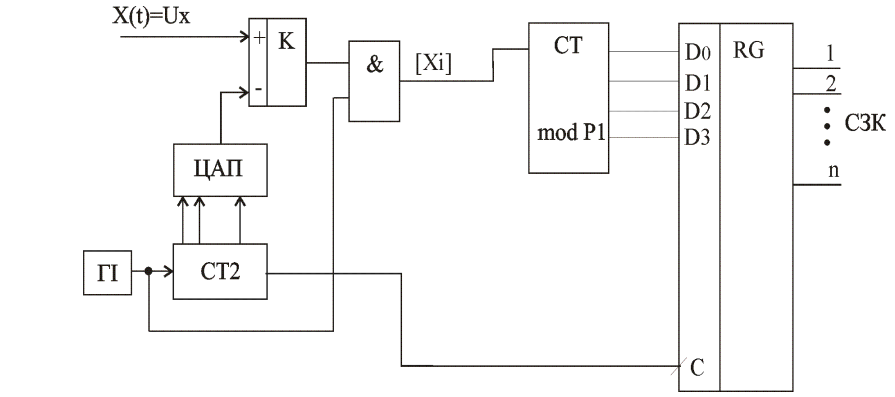


Рис.12.14. Структурна схема кодера по модулю *P1*.

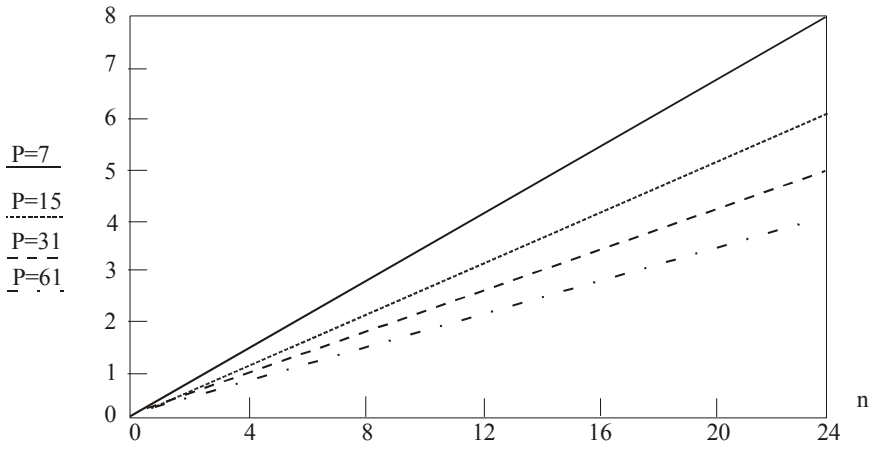


Рис.12.15. Коефіцієнт стиснення даних в залежності від модуля *P*.

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.

12.4. Кодування цифрових даних у базисі Галуа на основі вертикальної інформаційної технології.

Методи інтегрально-імпульсного кодування (ІІК) даних базуються на використанні вертикальної інформаційної технології в базисі Галуа.

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:

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0, \\ 0 & 1 & 0 & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 1 & 1 & 1 & 1 \end{array}$$

базису Радемахера замінюється рекурентним біт - орієнтованим кодом базису Галуа:

$$[0\,0\,0\,0]1\,0\,1\,0\,0\,1\,1\,0\,1\,1\,1\,1.$$

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

Вертикальна інформаційна технологія використана 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.12.16);

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

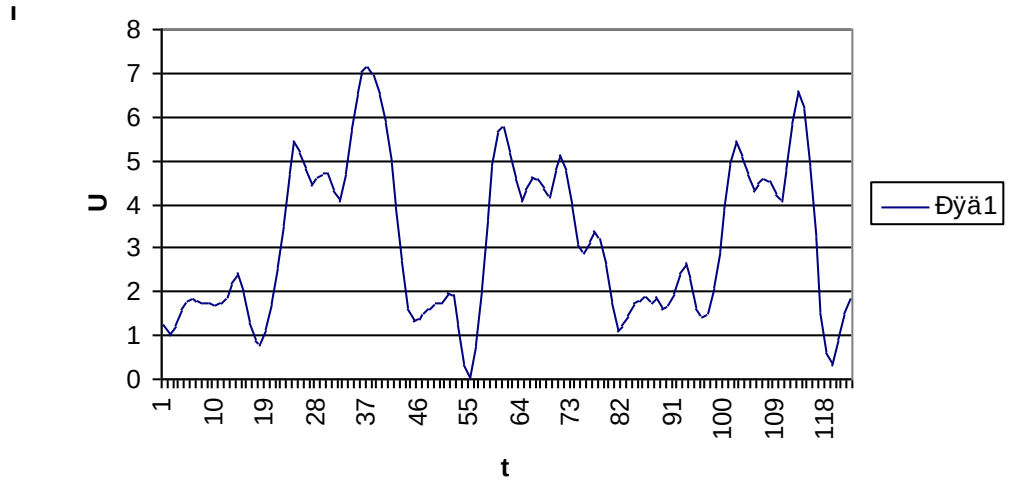


Рис.12.16. Графічне представлення виміряного технологічного параметру.

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.12.17);

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

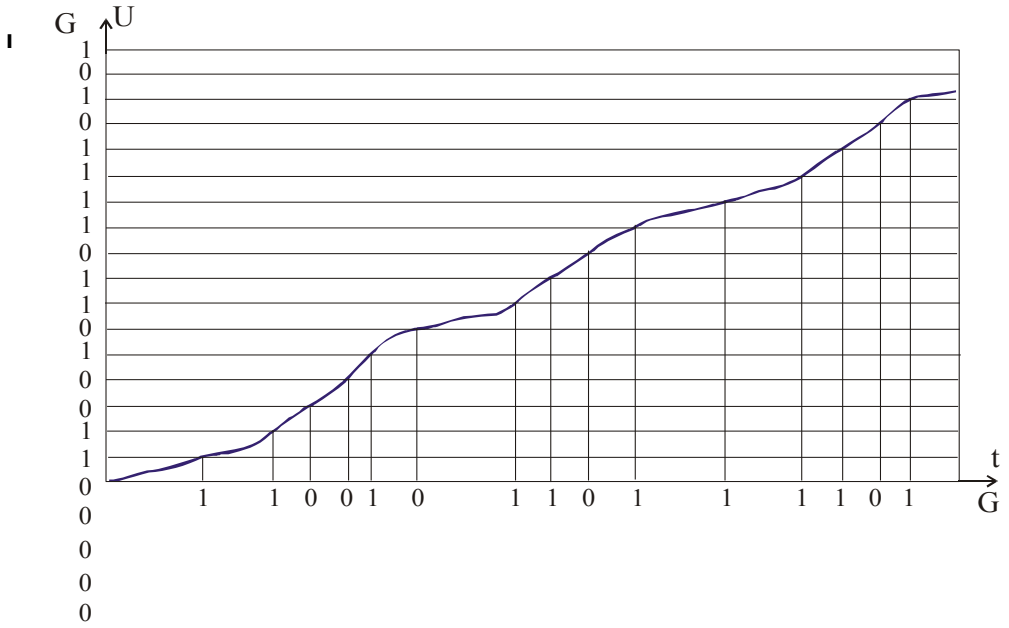


Рис. 12.17. Інтегральне представлення сигналу.

Суттєвим недоліком методу дельта-модуляції є спотворення інформації, як це показано на рис.12.18, при наявності різких змін вхідної послідовності відліків X_i, а також втрата абсолютного значення сигналу при втраті або спотворенні одного біту.

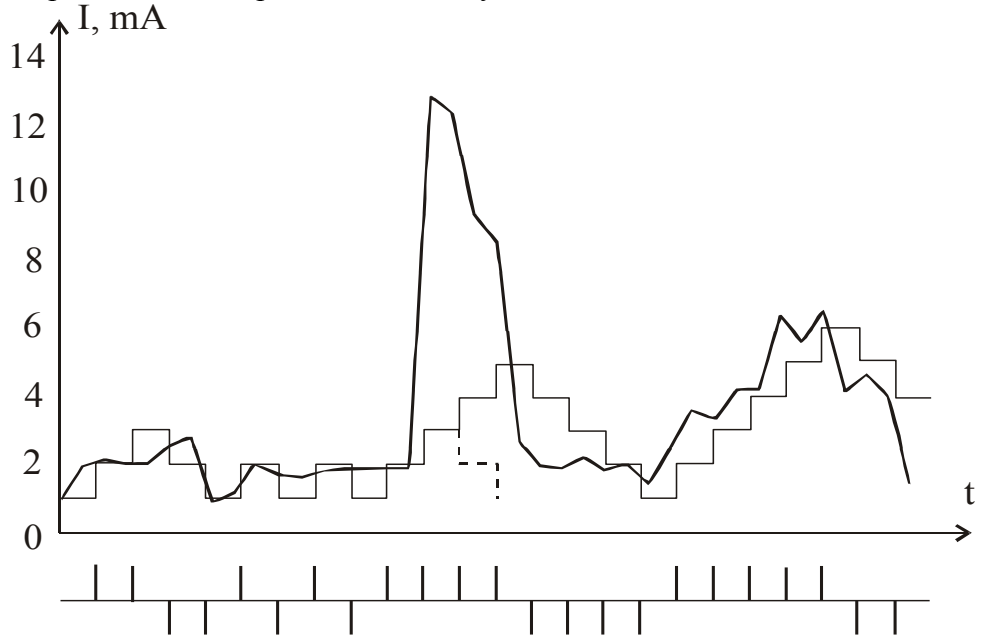
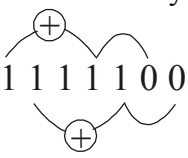


Рис.12.18. Дельта-модульований сигнал.

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.12.19 сигналу використовуємо 32– бітну послідовність Галуа:



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

При наявності стрибків сигналу (рис.12.18) інтеграл також різко зростає (рис.12.19), відповідно $\Delta t \to 0$.

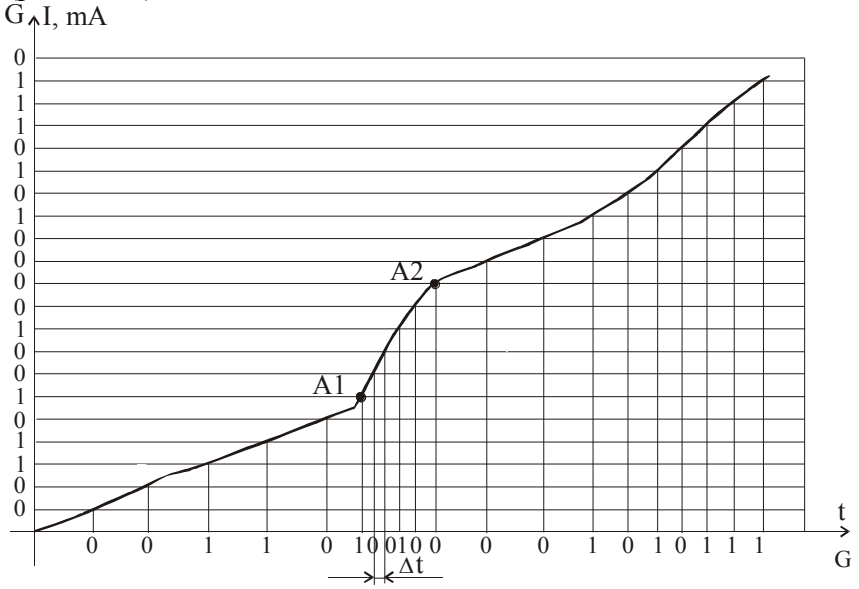


Рис.12.19. Інтегральне представлення сигналу.

Для визначення абсолютного значення сигналу (при втраті декількох бітів, область А1 – А2) (рис.12.19) 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.

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 $GF(p)$ є $\{0, 1, 2, \ldots, p-1\}$, а операції "+", "-", "х", ":" виконуються по модулю p [50 – 58].

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

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

При використанні примітивних незвідних 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 $\pi(x)$.

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 незвідного двійкового тричлена степені r, то перші r степенів елемента α представляють собою ефективний базис для запису поля $GF(2^r)$, так як множення на α може бути виконано з допомогою r – розрядного регістра, в зворотній зв'язок якого входить один суматор з двома входами (рис. 12.20).

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

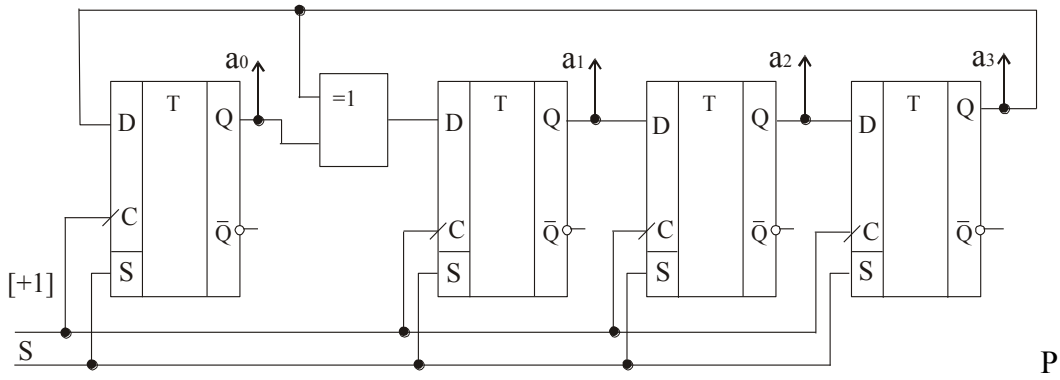


Рис.12.20. Формувач елементів поля Галуа $GF(2^4)$.

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:

а) в вигляді десяткових номерів (модифікованих логарифмів Log);

б) степенів α^i примітивного елемента α;

в) логарифмів $\log\left(\alpha^i\right)$ з основою α;

г) двійкових векторів;

д) поліномів;

е) розкладання по нормальному базису та інші.

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.12.20), використовуючи примітивний многочлен $g(x)=x^4+x+1$. Клас залишків $\{x\}=\alpha$, де α – 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 α в формі приведеній в таблиці 12.2.

Таблиця 12.2.

Представлення поля $GF(2^4)$.

Полярні координати			Прямокутні координати	
Десятковий номер N	Степінь α^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	α^0	0	0001	1
2	α^1	1	0010	x
3	α^2	2	0100	x^2
4	α^3	3	1000	x^3
5	α^4	4	0011	$x+1$
6	α^5	5	0110	x^2+x
7	α^6	6	1100	x^3+x^2

продовження таблиці 12.2.

8	α^7	7	1011	x^3+x+1
9	α^8	8	0101	x^2+1
10	α^9	9	1010	x^3+x
11	α^{10}	10	0111	x^2+x+1
12	α^{11}	11	1110	x^3+x^2+x
13	α^{12}	12	1111	x^3+x^2+x+1
14	α^{13}	13	1101	x^3+x^2+1
15	α^{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 $r=4$:

$x^3 \quad x^2 \quad x^1 \quad x^0$

$1 \quad 1 \quad 1 \quad 1 \quad \leftrightarrow x^3+x^2+x+1$

$1 \quad 0 \quad 1 \quad 1 \quad \leftrightarrow x^3+x+1$

Схему формування елементів поля $GF(2^4)$ в зворотному порядку отримаємо, змінивши підключення суматора по mod 2 (рис. 12.21).

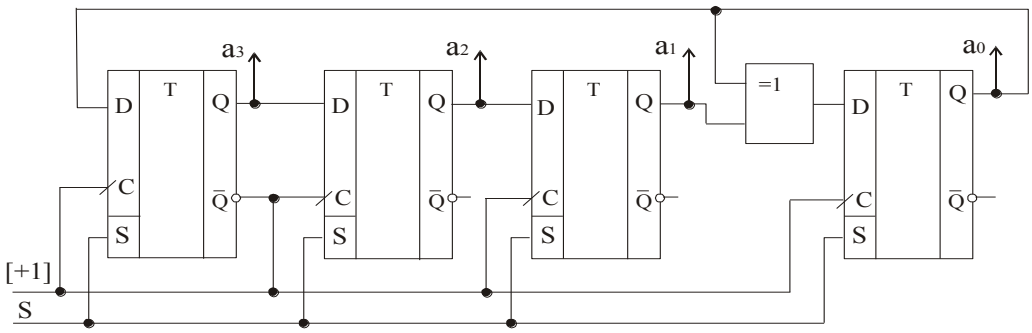


Рис.12.21. Схема формування елементів поля $GF(2^4)$ в зворотному порядку.

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.

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 $1, \alpha, \alpha^2, .. a^{r-1}$ при умові $a^r=1$.

Рекурентні співвідношення або різницеві рівняння:

$$\sum_{j=0}^{k} h_j \cdot \alpha_{i+j} = 0 \qquad (12.3)$$

або

$$\alpha_{i+k} = -\sum_{j=0}^{k-1} h_j \cdot \alpha_{i+j}, \qquad (12.4)$$

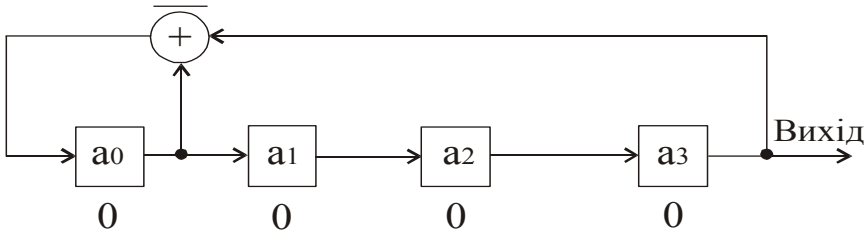
де $h_0 \neq 0$, $h_k = 1$ і кожне h_j належить полю $GF(q)$.

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

Співвідношення (12.4) визначає правило обчислення α_k по заданих значеннях величин $\alpha_0, \alpha_1, \cdots \alpha_{k-1}$. По відомих значеннях $\alpha_0, \alpha_1, \cdots \alpha_k$ можна знайти α_{k+1} і т.д.

Для обчислення суми (12.4) і, відповідно, для обчислення величин α_k по значеннях k попередніх членів послідовності використовується лінійна послідовна перемикаюча схема. Вихідні величини $\alpha_0, \alpha_1, \cdots \alpha_{k-1}$ заносимо в розряди пристрою, послідовні зсуви якого відповідають обчисленню послідовних символів, вихід після і-го зсуву дорівнює α_i.

На рис.12.22 показано регістр зсуву, лінійні зворотні зв'язки якого описуються непривідним примітивним багаточленом $h(x) = x^4 + x + 1$ над полем $GF(2)$.



Десяткове значення	Вихідний стан регістру			
	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.12.22. Схема формування та елементи поля $GF(2^4)$.

Вихід зчитується з правого кінця регістра 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

$$a = a_0\ a_1 \cdots\cdots = 1000100110\ \ 10111\,|\,1000 \cdots.$$

Недоліком представлених схем є необхідність завантаження початкового стану регістру.

Даний недолік можна усунути, якщо замість операції "виключаюче АБО" використати операцію "виключаюче АБО інвертоване" (рис.12.22).

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

0 0 0 0 1 1 1 0 1 1 0 0 1 0 1 | 0 0 0 0 .

Приведеною послідовністю можна закодувати десяткові числа від 0 до 14. Згідно з рекурентним співвідношенням (12.4) 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 незвідний примітивний поліном $h(x) = x^{20} + x^3 + 1$.

12.5. 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; ω – частота; t – час.

Побудуємо на цьому ж графіку (рис.12.23) 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.12.23) залежно від параметру t:

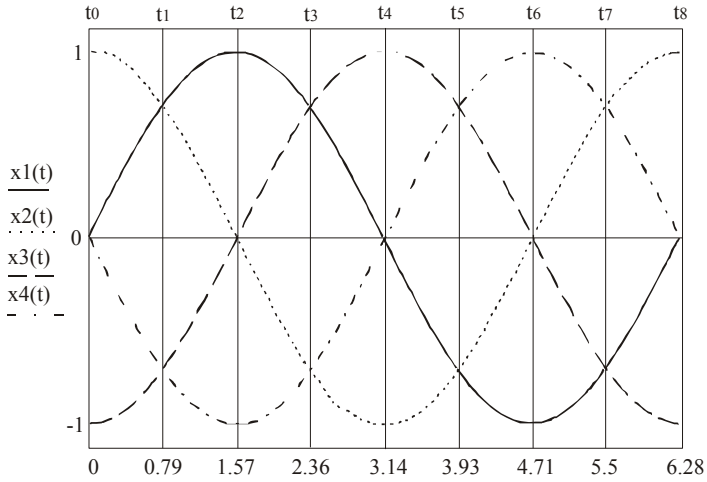


Рис.12.23. Чотирифазний гармонічний сигнал.

$$
\begin{aligned}
&t=0, \quad x_2(t)=A_0\\
&t=1, \quad x_1(t)=x_2(t)\\
&t=2, \quad x_1(t)=A_0\\
&t=3, \quad x_1(t)=x_3(t)\\
&t=4, \quad x_3(t)=A_0\\
&t=5, \quad x_3(t)=x_4(t)\\
&t=6, \quad x_4(t)=A_0\\
&t=7, \quad x_4(t)=x_2(t)
\end{aligned}
\qquad (12.5)
$$

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

При подачі на вхід порогових систем (рис.12.24) вхідного синусоїдального сигналу з довільною фазою і чотирифазним пороговим приймачем, описаного системою рівнянь (12.5), на виході отримаємо послідовність імпульсів (рис.12.25).

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:

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

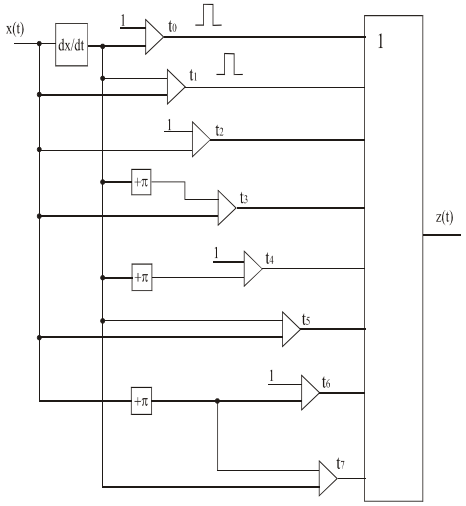


Рис.12.24. Порогова схема:

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

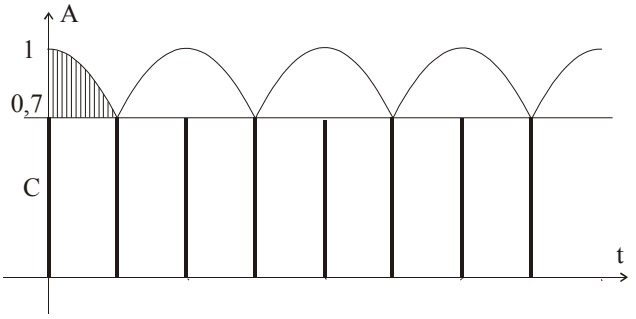


Рис. 12.25. Реакція порогової схеми на вхідний синусоїдальний сигнал.

В результаті, всі ці функції можуть бути зведені до одного фрагменту (рис.12.26), за допомогою якого можна однозначно відновити сигнал.

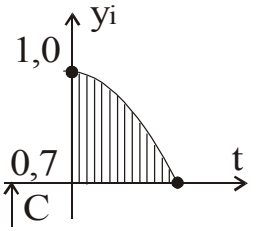


Рис.12.26. Фрагмент сигналу: C=0,707.

Декодування гармонічного сигналу відбувається на основі алгоритму (рис.12.27):

$$x(t)=\begin{cases} t_0 \le t < t_1, & f_0(t)=\int\limits_0^t (z_0(t)+C)dt; \\ t_1 \le t < t_2, & f_1(t)=\overrightarrow{C+z_0(t)}\,; \\ t_2 \le t < t_3, & f_2(t)=C+z_0(t); \\ t_3 \le t < t_4, & f_3(t)=\dfrac{d\overrightarrow{(z_0(t)+C}}{dt}; \\ t_4 \le t < t_5, & f_4(t)=\dfrac{d(z_0(t)+C)}{dt}; \\ t_5 \le t < t_6, & f_5(t)=0-\left(\overrightarrow{(z_0(t)+C)}\right); \\ t_6 \le t < t_7, & f_6(t)=0-\left((z_0(t)+C)\right); \\ t_7 \le t < t_8, & f_7(t)=0-\left(\int\limits_0^t \overrightarrow{(z_0(t)+C)}dt\right). \end{cases} \qquad (12.6)$$

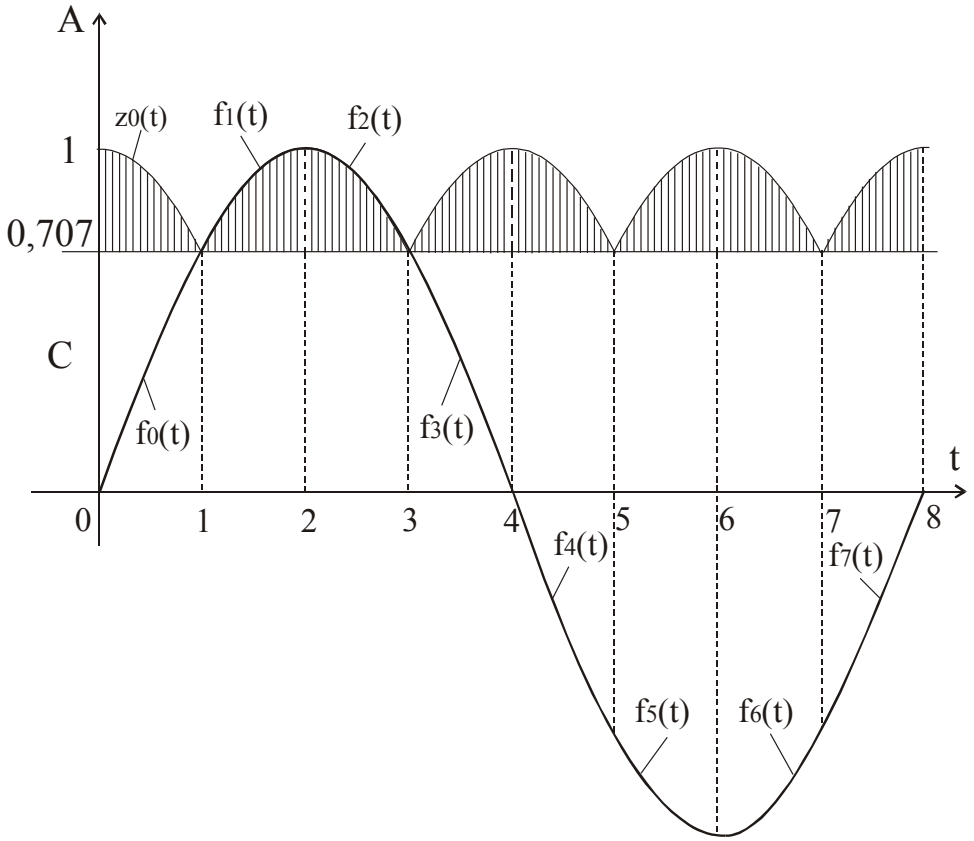


Рис.12.27. Декодування синусоїди на основі фрагменту $z_0(t)$.

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 (12.5).

На основі системи рівнянь (12.6) 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 $x(t)$;

Т – 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 $t_1 = t_2 = t_3 = ... = t_8$.

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 (12.6). 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 прямий 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.12.28) 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:

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

Перший прямий біт Галуа G вказує на те, що далі слідує масив значень кодового фрагменту Z_i, за допомогою якого відбувається декодування сигналу. Наступні прямі 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.12.28 і рис.12.29), описують реальний перехідний процес, що відбувається під час

аварійних станів на об'єктах електроенергетики. Початок перехідного процесу представлено на рис.12.28, а момент стабілізації перехідного процесу і переходу до нормального режиму роботи на рис. 12.29.

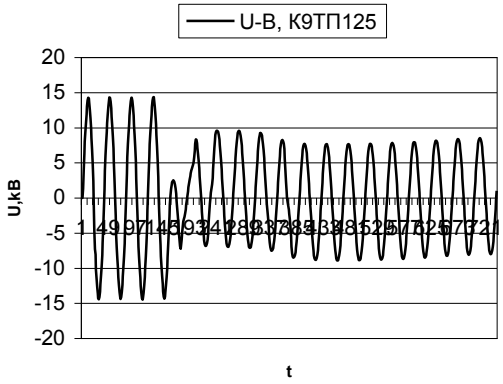


Рис.12.28. Осцилограма сигналу в момент аварії на об'єкті електроенергетики.

В результаті кодування сигналу представленого на рис.12.28 отримаємо потік даних:

$$G\{Z_i\}GGG\ldots G\,\overline{G}\{y_i\}\,\overline{G}\,\{y_i\}\ldots G\,\{Z_i\}GG\ldots\overline{G}\,\{y_i\}.$$

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:

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

де t_r – час реєстрації.

Об'єм даних для кодування перехідного процесу (рис.12.28) 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, $c_2 = 96$.

$I_\Sigma = 6076$ біт.

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

Коефіцієнт стиснення дорівнює:

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

При кодуванні сигналу, представленого на рис.12.29, отримаємо потік даних:

$$G\{Z_i\}\,GGGGGG\ldots GGGGG\ldots GGGGG,$$

його об’єм обчислюється за формулою

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

при $c_1 = 1$.

Коефіцієнт стиснення дорівнює:

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

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

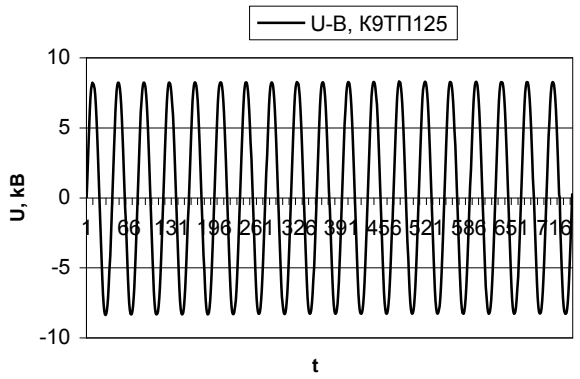


Рис.12.29. Осцилограма сигналу після аварії на об’єкті електроенергетики.

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.

12.6. Адаптивне стиснення одномірної інформації у базисі Галуа.

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.29.30.

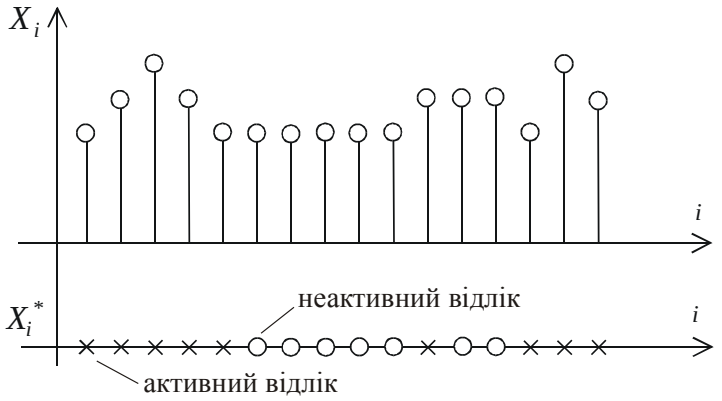


Рис.12.30. Решітчаста функція та послідовність активних і неактивних відліків: де $x_i = x_i^*$, $x_i \neq x_i - 1$, x_i^* – активний відлік, x_i – неактивний відлік.

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

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

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

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

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 $\overline{G}$ і після нього слідує значення контрольованого параметра, прямий біт Галуа G означає, що відлік неактивний.

Потік інформаційних даних набуде вигляду:

$$\overline{G}\,X^{*}GGG...\overline{G}\,X^{*}\overline{G}\,X^{*}GGG...G.$$

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

$$1 \le f \le m.$$

Можливі два варіанти слідування активних відліків f:

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

$$GGGGGG\,\overline{G}X^{*}GGGGGGGGG\;\overline{G}X^{*}GGGGGGGGG\;\overline{G}X^{*}GG.$$

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

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

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:

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

Обчислимо об'єм даних при різних методах кодування:

1) метод однопараметричного адаптивного кодування:

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

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]:

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

– $I_{3\min} = \hat{E}[\log_2 A] + \hat{E}[\log_2 m] + (f-1)\cdot\left(\hat{E}[\log_2 A]+1\right) =$ – 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. 12.31):

– 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 слідують 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.

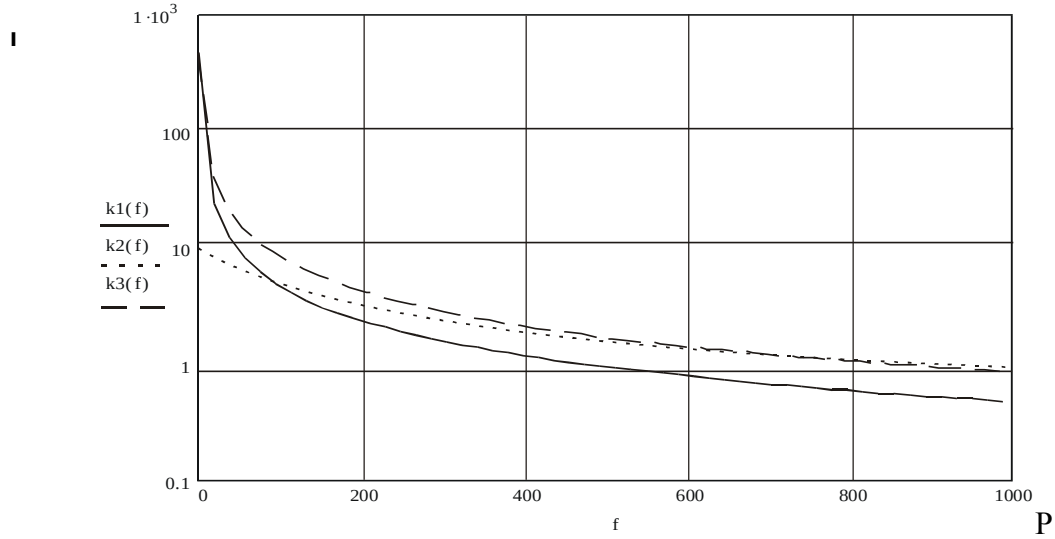


Рис.12.31. Залежності коефіцієнта стиснення від кількості активних відліків.

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.

12.7. Стиснення інформації в багатоканальних системах на основі вертикальної інформаційної технології у базисі Галуа.

Однією зі складних задач на рівні розподілених комп'ютерних систем (РКС) є впровадження стандартних протоколів зв'язку між сенсорами технологічних параметрів, контролерами і ПК. Високий рівень промислових завад, 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.

Властивості кодів Галуа, які визначають логічний зв'язок між різними елементами послідовності, відкривають нові можливості розробки

методів стиснення даних та організації блокової і символьної синхронізації в послідовних біт- та байт-орієнтованих потоках даних.

В багатоканальних системах контролю та управління на основі ВІТ (рис.12.32) дані в каналах змінюються в різні моменти часу, що ускладнює використання стандартних комунікаційних пристроїв.

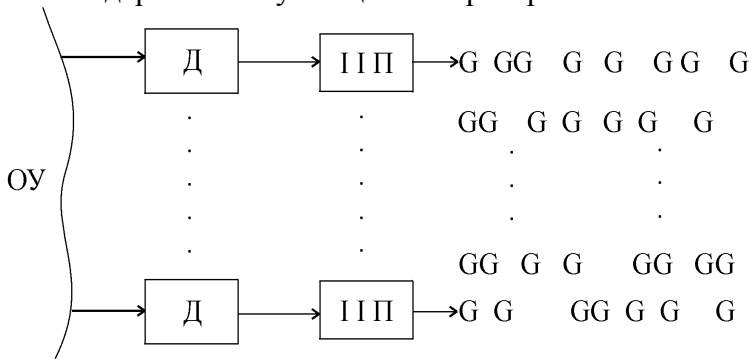


Рис.12.32. Кодування даних в багатоканальних системах на основі вертикальної інформаційної технології: Д – давач; ІІП – інтегрально імпульсний перетворювач.

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:

$$\begin{vmatrix} N & & & & \\ 1 & X^*_{i,1} & X_{i+1,1} & \dots & X^*_{m1} \\ 2 & X^*_{i,2} & X_{i+1,2} & \dots & X_{m2} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ j & X^*_{i,j} & X_{i+1,3} & \dots & X_{mj} \\ \dots & \dots & \dots & & \\ \dots & \dots & \dots & & \\ n & X_{i,n} & X^*_{i+1,n} & \dots & X^*_{mn} \end{vmatrix},$$

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 $\overline{G}$ – якщо відлік активний.

В результаті такого кодування формується наступний потік даних:

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

де СС – символи блокової синхронізації.

Коефіцієнт стиснення для запропонованого методу визначається за формулою:

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

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 $1 \le f \le 124$ представлено на рис.12.33.

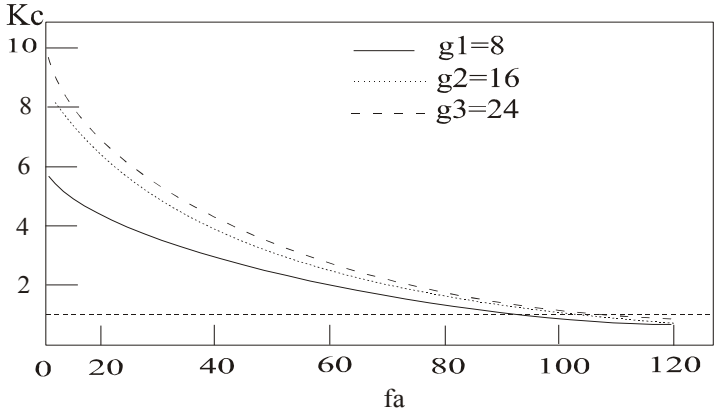


Рис.12.33. Залежність стиснення даних від кількості активних відліків.

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.

12.8. Кодування багатомірних джерел інформації у базисі Крестенсона-Галуа.

Кодування багатомірних ДІ є складною теоретичною та актуальною технічною задачею.

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

заданий вектор багатомірного простору N_k у вигляді

$$N_k = \{b_1, b_2, \ldots b_i, \ldots b_k\}, \tag{12.7}$$

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:

$$\{a^0, a^1, a^2, \ldots a^n\}, \tag{12.8}$$

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 $b_i << a^i$.

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:

$$b_i \le P_i - 1.$$

В даному випадку, застосувавши пряме перетворення системи залишкових класів над послідовністю (12.7), отримаємо:

$$N_k = res \sum_{i=1}^{n} B_i \cdot b_i (\text{mod } \rho),$$

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

$$B_i = \frac{\rho}{P_i} \cdot m_i \equiv 1 (\mathrm{mod}\ P_i),$$

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

Приклад кодування тривимірного джерела в базисі Крестенсона показаний на рис.12.34.

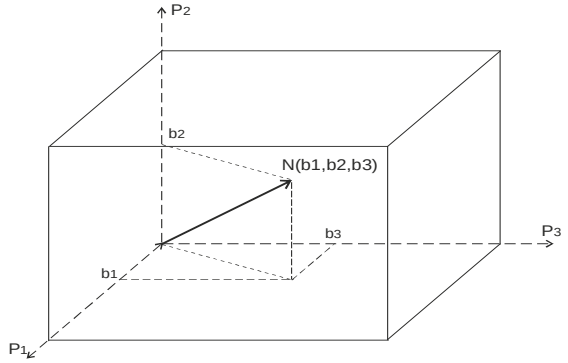


Рис.12.34. Кодування тривимірного джерела інформації в базисі Крестенсона-Галуа.

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.12.35.

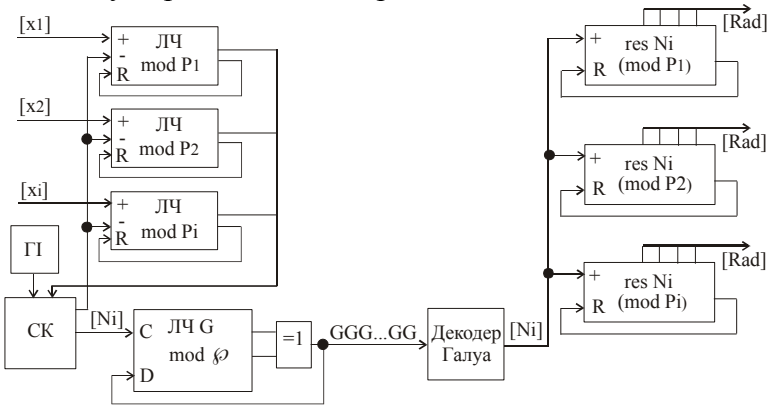


Рис.12.35. Структурна схема системи кодування та передавання даних у базисі Крестенсона-Галуа:

ЛЧ mod*Pi* – лічильник по модулю Pi; 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 ρ. 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.12.35).

Коефіцієнт зменшення надлишковості даних розраховують за формулою:

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

Важливою задачею є кодування багатомірних джерел інформації, які представляють багатовимірні вектори в вузлах двовимірної площини (рис.12.36).

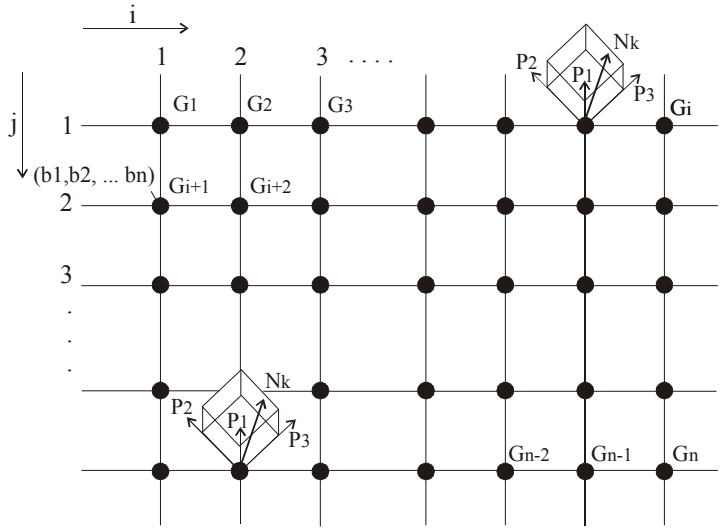


Рис.12.36. Модель джерела інформації.

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:

$$N_{k\,i\,j} = res\left(\sum_{i=1}^{k+2} b_i \cdot \beta_i \,(\mathrm{mod}\,\rho^*)\right),$$

де ρ^* – розширений модуль, який включає систему модулів

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

де $\begin{matrix} 0 < P_{k+1} \le \alpha - 1 \\ 0 < P_{k+2} \le \beta - 1 \end{matrix}$, $i = 1,2,\ldots,\alpha$, $j = 1,2,\ldots,\beta$.

Коефіцієнт стиснення для даного методу розраховується за формулою:

$$K_{c1} = \frac{\sum_{i=1}^{k} \hat{E}\left[\log_2 b_{i\max}\right] + \hat{E}\left[\log_2 \alpha\right] + \hat{E}\left[\log_2 \beta\right]}{\hat{E}\left[\log_2 (\rho^* - 1)\right]}. \qquad (12.9)$$

При застосуванні кодів Галуа вектор N_k доповнюється одним бітом Галуа, і двовимірне джерело представляється у вигляді одновимірного масиву даних

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

де G_l –рекурентна послідовність бітів Галуа,

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

Коефіцієнт стиснення при цьому визначається за формулою

$$K_{c2} = \frac{\sum_{i=1}^{k} \hat{E}\left[\log_2 b_{i\max}\right] + \hat{E}\left[\log_2 \alpha\right] + \hat{E}\left[\log_2 \beta\right]}{1 + \hat{E}\left[\log_2\left(\prod_{i=1}^{n} P_i - 1\right)\right]}. \qquad (12.10)$$

Графічна залежність коефіцієнта стиснення на основі залежностей (12.9) і (12.10) при $P1=17,\ P2=53,\ P3=153,\ \alpha=10,\ \beta=12$ подана на рис.12.37.

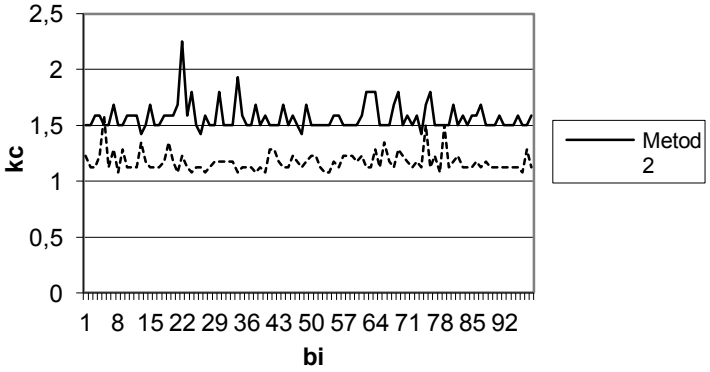


Рис.12.37. Залежність коефіцієнта стиснення від значення виміряних параметрів.

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

12.9. Стиснення алфавітно-цифрової інформації у базисі Галуа.

Стиснення інформації застосовується для прискорення та зниження витрат на її оброблення, зберігання й пошук, а також для зменшення пам'яті, зайнятої в ЕОМ.

Під стисненням інформації розуміється операція, внаслідок якої певному коду чи повідомленню ставиться у відповідність код або повідомлення меншої довжини

Існує багато практичних алгоритмів стиснення даних. Однак, в основі цих методів лежать три теоретичних алгоритми: алгоритм RLE (Run Length Encoding); алгоритми групи KWE (KeyWord Encoding); алгоритм Хафмана. В основі алгоритму *RLE* лежить ідея виявлення послідовностей даних, що повторюються, та заміни цих послідовностей більш простою структурою, в якій вказується код даних та коефіцієнт повторення. В основі *алгоритму KWE* покладено принцип кодування лексичних одиниць групами байт фіксованої довжини. Результат кодування зводиться в таблицю, утворюючи так званий словник. В основі алгоритму Хафмана лежить ідея кодування бітовими групами. Після частотного аналізу вхідної послідовності символи

сортуються за спаданням частоти входження. Чим частіше зустрічається символ, тим меншою кількістю біт він кодується. Результат кодування зводиться в словник, що необхідний для декодування [I].

Застосування базису Галуа для стиснення алфавітно-цифрової інформації виконується згідно наступної формули: $G_{j+i} = G_i \oplus G_{i-n}$ (табл.12.3.)

Таблиця 12.3.

Приклад коду Галуа з об'ємом коду V=N:

1	2	3	4	5	6	7	8
0	0	0	1	0	1	1	1

Використання восьмибітового слова дає змогу закодувати 2=256 знаків, тоді як реальні алфавіти з урахуванням цифр і деяких допоміжних символів містять до 50-60 знаків, тобто для кодування їх потрібні п'яти-шестибітові комбінації та аналогічні структури пам'яті. Спробуємо зменшити кількість біт, необхідних для кодування одного символу, поділивши всі символи на чотири групи, що показані в табл.12.4.

Табляця 12.4.

Кодова таблиця Галуа з поділом на групи.

1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1
				X	t	h	e	S	i	п	d	0	г	a	m	P	1	У	c	U	ь	j	f	g	W	k	V	q	z		-
				д	к	а	й	м	0	в	і	р	н	я	с	т	У	е	ш	ц	л	г	ч	З	Х	п	и	ь	б		-
				ф	Щ	ю	ж	ъ	Ы	Э	є	ї	ё	=	+	-	0	1	2	3	4	5	6	7	8	9	*	/	A		-
				ч			?	і	U̇	№	%	@	(	)	~	'	U	#	$	&	1	\	<	>	{	}	[	]			-

Останні три біти послідовності Галуа визначають номер групи, в якій знаходиться заданий символ:

00011 - друга група; 00111 - третя група; 01111 - четверта група.

Якщо ми знаходимось в певній групі, а хочемо закодувати символ латинського алфавіту, потрібно ще раз вказати код заданої групи. Для кодування великої літери будемо використовувати послідовність 00000. Позначимо кожен символ 5 бітами Галуа, Кожен наступний символ позначимо бітами попереднього, зсунутого вліво на 1 біт. Комбінацію тих літер, що зустрічаються часто, розташуємо поруч. Тепер замість кожного символу тексту запишемо відповідні 5 біт коду. Якщо наступний символ

тексту відповідає наступному символу кодової таблиці, то запишемо відповідний біт, в іншому випадку - інвертований біт Галуа.

Спробуємо даний метод для кодування символів з різних груп. Так, слово Windows98 буде зашифровано наступним чином:

W	G	і	п	d	0	G	W	G	в	G	III група	9	G	8
00000 01110	0	00110	1	0	0	0	01110	0	10011	1	00111	11101	0	01110

Дане слово займає 72 біт пам'яті. Після упаковки його за допомогою 5-бітової послідовності Галуа воно займатиме 48 біт. Отже Кс=1,5.

Мінімальний коефіцієнт стиснення даних, тобто при умові, що в тексті немає жодної пари символів, які б в кодовій таблиці Галуа знаходилися поруч буде розраховуватися за формулою:

$$K_{c\min} = \frac{8 \cdot n}{(6 \cdot n + 5 \cdot n_v + 5 \cdot n_r)} \qquad (12.11)$$

де n - кількість символів в файлі, що кодується, n_v — кількість великих символів, n_r - кількість переходів з однієї групи кодової таблиці Галуа в іншу.

Якщо ж враховувати, що в більшості текстів будуть зустрічатися пари символів, які в кодовій таблиці Галуа знаходяться поруч, тоді попередня формула матиме вигляд:

$$K_{c\min} = \frac{8 \cdot n}{(6 \cdot n - 5 \cdot n_p + 5 \cdot n_v + 5 \cdot n_r)}$$

де n_p - кількість символів, які в таблиці Галуа знаходяться поруч.

Для того, що запропонований метод стиснення був більш ефективним, необхідно враховувати статистичні характеристики алфавіту, з якого складаються тексти, які слід обробляти. Бажано також враховувати ймовірність різних сполучень деяких знаків.

Розглянемо приклад застосування базису Галуа для стиснення алфавітно-цифрової інформації.

Нехай заданий наступний текст:

Теорема множення ймовірностей: ймовірність добутку двох подій дорівнює ймовірності однієї з них, помноженій на умовну ймовірність другої при наявності першої. Теорема додавання ймовірностей: ймовірність суми двох подій дорівнює сумі ймовірностей події мінус ймовірність їх добутку.

Проаналізуємо частоту появи складів у цьому тексті. Найчастіше зустрічаються сполучення літер те, ня, вір, ймо, су, ює, до, єї.

Необхідно 255 біт Галуа, щоб позначити всі літери, цифри та символи, що можуть зустрічатися в текстах. Тому прогенеруємо послідовність Галуа з ключем 1 0 0 0 1 1 1 0. Отримаємо:

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

Позначимо кожен символ 8 бітами Галуа. Кожен наступний символ будемо позначати бітами попереднього, зсунутого вліво на 1 біт. Комбінацію тих літер, що зустрічаються часто, розташуємо поруч. Тепер замість кожного символу тексту запишемо відповідні 8 біт коду. Якщо наступний символ тексту відповідає наступному символу кодової таблиці, то запишемо відповідний біт, в іншому випадку – інвертований біт Галуа. Розглянемо фрагмент кодової таблиці:

Caps Lock								к	а	й	м	о	в	і	р	н	я	с	у	т	е	д	о	є	ї	л	ь	
1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	0	1	0	1

Слово “ймовірність” буде зашифроване наступним чином:

й								м	о	в	і	р	н	$\overline{G}$	і							$\overline{G}$
1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0
с								$\overline{G}$	Т							$\overline{G}$						
0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0					
ь								$\overline{G}$														
1	1	1	1	1	0	1	0	0														

Дане слово займає 88 біт пам’яті. Після упаковки його за допомогою послідовності Галуа воно займатиме 51 біт. Отже, Кс=і,73.

Спробуємо упакувати заданий текст згідно з методом і оцінити коефіцієнт стиснення

До моменту стиснення текст займав 2354 біт. Після стиснення - 1929.

Коефіцієнт стиснення дорівнює:

Кс=2354/1929=1,22

Якщо спробувати упакувати цей самий текст за допомогою архіватора PKZIP, то він займатиме 2216 біт, тобто Кс=1,06. Як бачимо, даний метод є доволі ефективним для стиснення текстових даних.

Проведемо деякі дослідження залежності коефіцієнта стиснення від зміни параметрів.

Нехай, задана послідовність чисел, що складається зі 100 символів. Оскільки всі числа, в тому числі й пробіл, належать до однієї групи, то значення змінних $n_v = n_r = 0$. А коефіцієнт стиснення, без врахування кількості розташованих поруч цифр буде рівний $K_c = \frac{8 \cdot n}{6 \cdot n} = \frac{8 \cdot 100}{6 \cdot 100} = 1{,}33$. Дана величина є сталою, незалежно від кількості символів у файлі.

Якщо ж одна п'ята частина символів заданої послідовності знаходиться поруч в кодовій таблиці, то коефіцієнт стиснення збільшиться до $K_c = \frac{8 \cdot n}{6 \cdot n - 5 \cdot n_p} = \frac{8 \cdot 100}{6 \cdot 100 - 5 \cdot 20} = 1{,}6$

Як бачимо, ефективність стиснення збільшилась.

Розглянемо, як буде змінюватися коефіцієнт стиснення $K_{c\min}$ при зміні параметрів. Якщо кількість великих літер та кількість зміни групи символів буде незмінною, а кількість символів в тексті буде зростати, то коефіцієнт стиснення буде збільшуватися. Проаналізувавши формулу (12.11) отримаємо, що ефективність стиснення буде досягнута, при умові $5 \cdot n_v + 5 \cdot n_r \le 2 \cdot n$. Як видно з рис.12.39, якщо $n_v = n_r = 10$, то *Кс1>1,* при *n>50.*

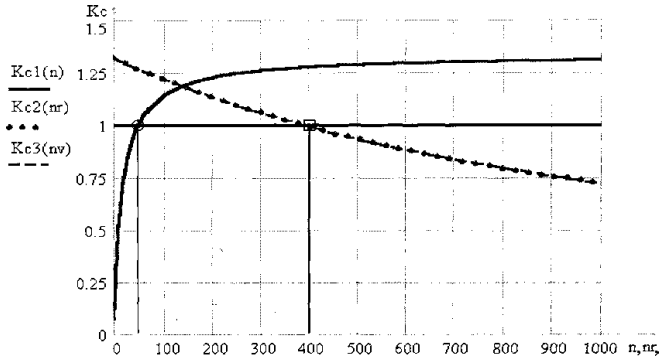


Рис.12.39. Залежність коефіцієнта стиснення від кількості символів в текстовому файлі (Ксі), від кількості змін групи символів (Кс2) та кількості великих літер (Кс3).

Коефіцієнти Кс2 та Кс3, що представлені на рис.12.39, показують залежність коефіцієнта стиснення від кількості змін групи символів та кількості великих літер. Як бачимо, вони рівні і оберненопропорційно залежать від заданих параметрів. Якщо *n=1000,* то *Кс1>1,* при n_v або n_r *<400.*

На рис.12.40 представлено залежність коефіцієнта стиснення від кількості символів при умові, що в тексті немає жодної пари символів, які б в кодовій таблиці Галуа знаходилися поруч (Кс1) та залежність коефіцієнта стиснення від кількості символів, які в таблиці Галуа знаходяться поруч (Кс2). Аналіз отриманих залежностей дозволяє зробити висновки: якщо Кс1>1 при умові $5 \cdot n_v + 5 \cdot n_r \le 2 \cdot n$ і зі збільшенням n наближається до величини 1,33, то Кс2>1 при умові $5 \cdot n_v + 5 \cdot n_r - 5 \cdot n_p \le 2 \cdot n$. Якщо ж кількість n_p досягає значення 0,5*n*, то ефективність стиснення збільшується в 2,25 раз.

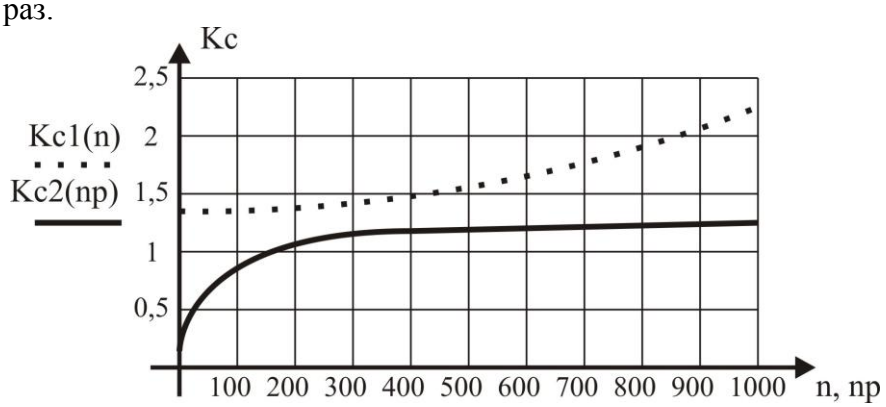


Рис.12.40. - Залежність коефіцієнта стиснення від кількості символів (Ксі) та від кількості символів, які в таблиці Галуа знаходяться поруч (Кс2).

Розглянемо залежність коефіцієнту стиснення від відношення загальної кількості символів до суми кількості змін групи символів та кількості великих літер (рис.12.41). Кс>1 при $\frac{n}{n_v - n_r} > 2{,}5$.

Отже, для одноалфавітних текстових даних з мінімальною кількістю великих літер та максимальною кількістю символів, розташованих поруч в кодовій таблиці даний метод дає найбільший ефект.

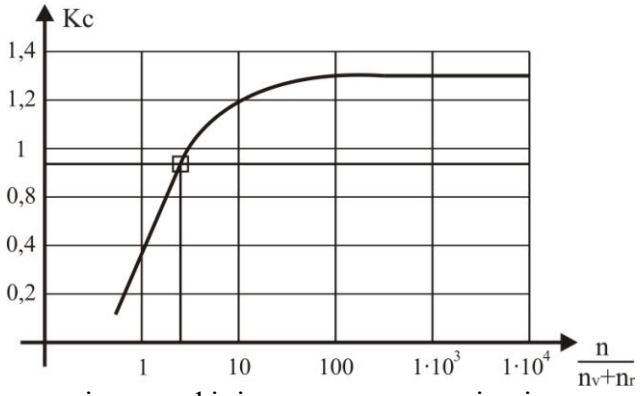


Рис.12.41. Залежність коефіцієнта стиснення від відношення кількості символів в текстовому файлі до суми кількості змін групи символів та кількості великих літер.

Описаний метод є більш ефективним для одно алфавітних з невеликою кількістю великих літер та максимальною кількістю символів, розташованих поруч у кодовій таблиці Галуа.

РОЗДІЛ 13

ПЕРЕДАВАННЯ СИГНАЛІВ ТА ІНФОРМАЦІЙНИХ ПОТОКІВ НА ОСНОВІ КОДІВ ПОЛЯ ГАЛУА

13.1. Інтегрально-імпульсна технологія формування знакозмінних даних в кодах поля Галуа.

Новим напрямком у розвитку обчислювальних систем та систем передавання даних є розробка інформаційної технології в базисі Галуа. Особливість цієї технології полягає у біт-орієнтованому представленні інформації, яке дозволяє відображати інкрементні відліки одним бітом після завантаження початкового n-розрядного коду , де n – розрядність даних в класичному двійковому представленні (в базисі Радемахера).

Між кодами в базисі Радемахера і кодами поля Галуа існує однозначна відповідність, в якій легко помітити, що інкрементні коди Галуа використовують в $n-1$ старших розрядах молодші $n-1$ біт попереднього коду, і тільки в молодшому розряді ставиться новий біт, який формується у відповідності до вибраного ключа:

10-ва система	2-ва система	Галуа система
0	00...0000	00...0000
1	00...0001	00...0001
2	00...0010	00...0010
3	00...0011	00...0101
4	00...0100	00...1011
5	00...0101	00...0110
6	00...0110	00...1100
⋮	⋮	⋮
254	11...1110	01...0000
255	11...1111	10000000

Таким чином, розрядність обчислювальних та приймально-передавальних пристроїв можна зменшити до одного біта.

Для досягнення інкрементної зміни відліків, які характеризують певний параметр процесу ОУ пропонується відображати параметр не миттєвим значенням, а інтегральним. Це дозволяє отримати зростаючі відліки при спаданні значень самого параметру. Перехід від миттєвих дискретних значень до інтегральних можна здійснити використавши числові методи:

$$\begin{cases} S_{\text{р‘}} = n \cdot \Delta t \sum_{i=0}^{n-1} x_i \ - \text{метод лівих прямокутників;} \\ S_{\text{р•}} = n \cdot \Delta t \sum_{i=1}^{n} x_i \ - \text{метод правих прямокутників;} \\ S_{\text{р}} = \dfrac{n \cdot \Delta t}{2} \sum_{i=0}^{n-1} \left(x_i + x_{i+1} \right) - \text{метод трапецій} \end{cases} \qquad (13.1)$$

та інші, де n – кількість дискретних відліків;

Δt – період дискретизації; x_i – дискретний відлік;

S – інтегральне значення параметру.

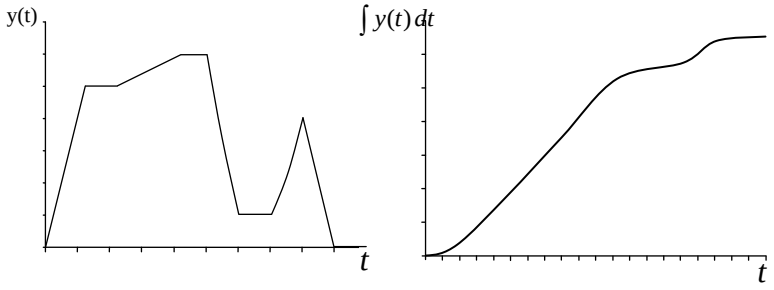
Для спрощення процедури інтегрування та повернення до миттєвих значень доцільно вибрати період дискретизації Δt рівний одиниці часу, яка забезпечує необхідну точність. Тоді вирази (3.11) матимуть вигляд:

$$S_{\text{р‘}} = \sum_{i=0}^{n-1} x_i \ ;$$

$$S_{\text{р•}} = \sum_{i=1}^{n} x_i \ ;$$

$$S_{\text{р}} = \frac{1}{2} \sum_{i=0}^{n-1} \left(x_i + x_{i+1} \right).$$

Інтегрально-імпульсне представлення станів ДІ найбільш ефективне в тому випадку, коли відліки параметру приймають тільки невід’ємні значення. Інакше необхідно вводити додаткові процедури для коректування кодів та отримання достовірних результатів. Це обумовлено тим, що при відсутності від’ємних відліків інтегральне значення параметру не буде мати спадаючих ділянок, і його можна відображати інкрементними кодами (рис.13.1).



а) б)

Рис. 13.1. а) - графік процесу , б) - інтегральне представлення.

Знакозмінні процеси в інтегральній інформаційній технології пропонується відображати трьома способами.

Перший спосіб полягає у використанні двохбітної реверсивно-кодової шкали Галуа. В цьому методі кожен дискретний відлік кодується двома бітами, де перший біт приймає значення прямого біта Галуа G при зростанні інтегрального значення на один квант, при цьому другому біту присвоюється інвертоване значення біту Галуа $\overline{G}$. Якщо інтегральне значення зменшується на один квант, то другий біт рівний G, а перший встановлюється в $\overline{G}$. Таким чином, перші біти складають пряму послідовність, яка відображає тільки зростання інтегрального значення S^+, а другі біти утворюють реверсивну послідовність, яка характеризує тільки спадання інтеграла параметра S^-. Тоді миттєве значення при такому способі кодування визначається різницею між інтегральними значеннями за виразом:

$$x_i = S_i^+ - S_{i-1}^+ + S_{i-1}^- - S_i^- .$$

Другий спосіб полягає у використанні подвійного інтегралу параметра процесу. Проте цей спосіб придатний для процесів, які змінюються за синусоїдальною функцією (рис. 13.2) або деякими іншими функціями, в яких перший інтеграл приймає невід'ємні значення.

Цей спосіб більш зручний у представленні в кодах поля Галуа, але має обмеження в застосуванні і складну процедуру отримання миттєвих значень з використанням різниць другого порядку. При обчисленні інтегральних значень методом лівих прямокутників отримати вихідні миттєві значення можна за виразом:

$$x_i = S_i^2 - 2S_{i-1}^2 + S_{i-2}^2 = S_i^2 - S_{i-1}^2 - S_{i-1}^1 ;$$

де: $S_i^1 = S_{i-1}^1 + x_i$, $S_0^1 = x_0$;

$S_i^2 = S_{i-1}^2 + S_i^1 = (i+1)\cdot x_0 + i \cdot x_1 + \ldots + 2x_{i-1} + x_i$;

$S_0^2 = S_0^1 = x_0$.

Третій спосіб є найбільш простим. Суть методу полягає у піднятті відліків сигналу до діапазону невід'ємних значень, шляхом додавання константи C, що вибирається рівною модулю найменшого від'ємного значення, яке може приймати сигнал (рис. 13.3).

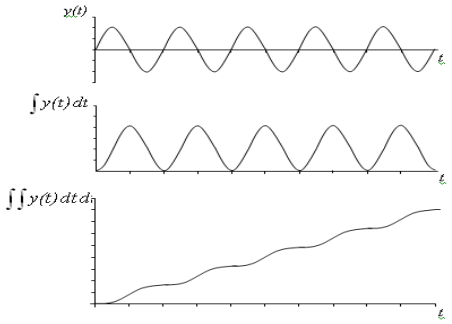


Рис.13.2. Знакозмінний процес та його представлення через перший і другий інтеграли.

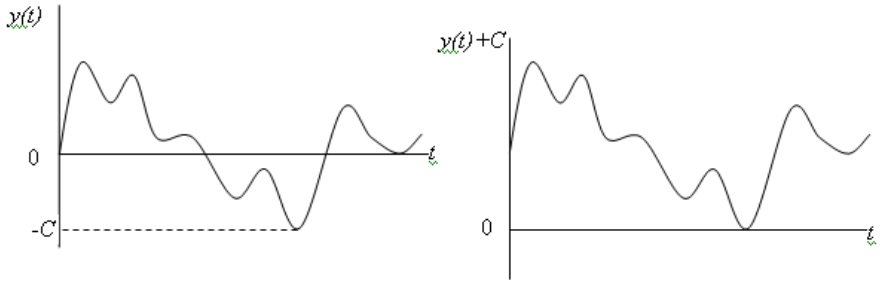


Рис.13.3. Приведення знакозмінного процесу до процесу з невід'ємними значеннями.

Таким чином, інтеграл параметру не буде мати спадаючих ділянок. При поверненні до миттєвих значень необхідно враховувати, що фізичні значення менші на константу C і в шкалі Галуа їм будуть відповідати зміщені коди.

Подані моделі дозволяють представити відліки одноканальних об'єктів керування одним (рис.13.4,(а) або двома (рис.13.4,(б) (при використанні реверсивно-кодової шкали Галуа) бітами.

Аналогічну модель в інтегральній технології можна побудувати для багатоканальних ДІ.

Найбільш зручний спосіб формування даних у вигляді кодів Галуа – асинхронний, який полягає в тому, що біти Галуа генеруються тільки в моменти часу зміни інтегрального стану ДІ на один квант. В результаті в кожному каналі ДІ формується послідовність бітів Галуа (рис.13.5.).

Для визначення інтегральних значень сигналу в дискретний момент i, в кожному каналі зчитуються останні n біт, де n – вибрана розрядність коду.

Для опрацювання даних в інтегрально-імпульсній технології розроблено комплекс ЛСІМ, адаптованих до представлення станів ДІ

інтегральними значеннями. Вони дають можливість фіксувати амплітудні, динамічні, фазові та спектральні відхилення станів ДІ, використовуючи при цьому значення інтегральних характеристик.

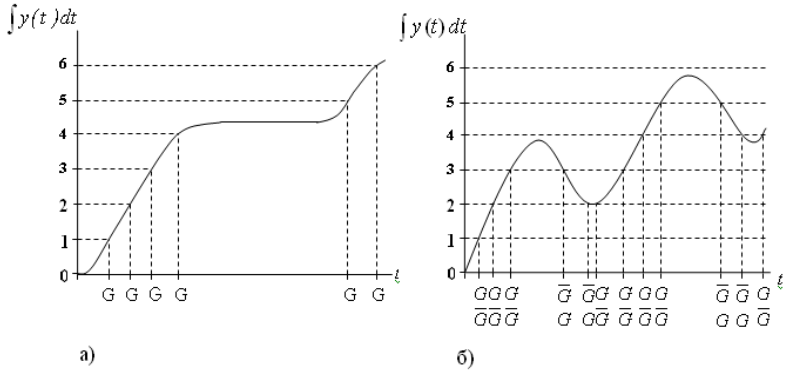


Рис. 13.4. Інформаційна модель ДІ в ІІТ.

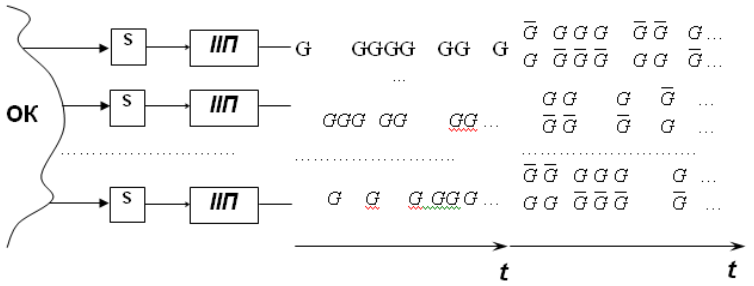


Рис.13.5. Кодування багатоканальних ДІ у базисі Галуа. S – сенсор; ІІП – інтегрально-імпульсний перетворювач.

Інтегрально-імпульсна технологія у базисі Галуа має ряд переваг в порівнянні з традиційними технологіями побудови інформаційно-обчислювальних систем, що використовуються в РКС. Основні переваги полягають в наступному:

— зменшується об’єм даних на низових рівнях РКС та необхідна швидкість їх передавання по каналах зв’язку;

— представлення даних в базисі Галуа забезпечує високу завадозахищеність інформації.

13.2. Особливі автокореляційні характеристики кодів Галуа.

Існують 2 типи цифрових кореляційних приймачів ШПС: автокореляційні та взаємокореляційні. Найчастіше для розрахунку автокореляційних та взаємокореляційних функцій ШКП використовується

мультиплікативна згортка знакових представлень шумоподібних сигналів $\mathrm{sign}(\overset{\circ}{x}_i)$. В той же час існує 8 типів кореляційних функцій, які характеризуються різними параметрами точності, алгоритмічної складності та структурної складності СП, які їх реалізують.

В табл.13.1. приведені аналітичні вирази цифрових авто- та взаємокореляційних функцій, що використовуються для реалізації спецпроцесорів приймання одновимірних ШКП.

В таблиці 13.1:

$$sign(\overset{\circ}{x}_i)=\begin{cases}+1, \overset{\circ}{x}_i \geq 0\\ 0, \overset{\circ}{x}_i = 0\\ -1, \overset{\circ}{x}_i < 0\end{cases}; \qquad sign(\overset{\circ}{y}_{i+j})=\begin{cases}+1, \overset{\circ}{y}_{i+j} \geq 0\\ 0, \overset{\circ}{y}_{i+j} = 0\\ -1, \overset{\circ}{y}_{i+j} < 0\end{cases}$$

$$\overset{\circ}{x}_i = x_i - M_x;\ \overset{\circ}{y}_i = y_i - M_y; \qquad M_x = \frac{1}{n}\sum_{i=1}^{n} x_i;\ M_y = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$D_x = \frac{1}{n}\sum_{i=1}^{n}(x_i - M_x)^2; \qquad D_y = \frac{1}{n}\sum_{i=1}^{n}(y_i - M_y)^2;$$

$$\breve{Z}_{xx}=\begin{cases}x_i, x_i < x_{i+j}\\ x_{i+j}, x_i \geq x_{i+j}\end{cases}; \qquad \breve{Z}_{xy}=\begin{cases}x_i, x_i < y_{i+j}\\ y_{i+j}, x_i \geq y_{i+j}\end{cases},$$

де $sign(\overset{\circ}{x}_i)$, $sign(\overset{\circ}{y}_{i+j})$ - знаки центрованих значень; M_x, M_y - математичні сподівання; D_x, D_y - дисперсії; $\breve{Z}_{xx}, \breve{Z}_{xy}$ - елементи кореляційної еквівалентності.

Таблиця 13.1.

Аналітичні вирази кореляційних функцій.

№	Кореляційна функція	Автокореляційна функція	Взаємокореляційна функція
1	знакова	$H_{xx}(j) = \frac{1}{n}\sum_{i=1}^{n} sign(\overset{\circ}{x}_i) \times sign(\overset{\circ}{x}_{i+j})$	$H_{xy}(j) = \frac{1}{n}\sum_{i=1}^{n} sign(\overset{\circ}{x}_i) \times sign(\overset{\circ}{y}_{i+j})$
2	релейна	$B_{xx}(j) = \frac{1}{n}\sum_{i=1}^{n} \overset{\circ}{x}_i \times sign(\overset{\circ}{x}_{i+j})$	$B_{xy}(j) = \frac{1}{n}\sum_{i=1}^{n} \overset{\circ}{x}_i \times sign(\overset{\circ}{y}_{i+j})$

продовження таблиці 13.1

3	коваріаційна	$K_{xx}(j)=\frac{1}{n}\sum_{i=1}^{n}x_i\times x_{i+j}$	$K_{xy}(j)=\frac{1}{n}\sum_{i=1}^{n}x_i\times y_{i+j}$
4	кореляційна	$R_{xx}(j)=\frac{1}{n}\sum_{i=1}^{n}\overset{\circ}{x}_i\times\overset{\circ}{x}_{i+j}$	$R_{xy}(j)=\frac{1}{n}\sum_{i=1}^{n}\overset{\circ}{x}_i\times\overset{\circ}{y}_{i+j}$
5	нормована кореляційна	$\rho_{xx}(j)=\frac{R_{xx}(j)}{D_x}$	$\rho_{xy}(j)=\frac{R_{xy}(j)}{\sqrt{D_x\times D_y}}$
6	структурна	$C_{xx}(j)=\frac{1}{n}\sum_{i=1}^{n}(x_i-x_{i+j})^2$	$C_{xy}(j)=\frac{1}{n}\sum_{i=1}^{n}(x_i-y_{i+j})^2$
7	модульна	$G_{xx}(j)=\frac{1}{n}\sum_{i=1}^{n}\lvert x_i-x_{i+j}\rvert$	$G_{xy}(j)=\frac{1}{n}\sum_{i=1}^{n}\lvert x_i-y_{i+j}\rvert$
8	еквівалентності	$F_{xx}(j)=\frac{1}{n}\sum_{i=1}^{n}\breve{Z}_{xx}$	$F_{xy}(j)=\frac{1}{n}\sum_{i=1}^{n}\breve{Z}_{xy}$

З табл.13.1 видно, що найменшою алгоритмічною складністю характеризуються функції еквівалентності $F_{xx,}(j)$ та $F_{xy}(j)$, в яких базисна операція згортки виконується шляхом порівняння модульних значень $x_i,\ldots,$ x_{i+j} та сумування менших з них.

Описані математичні рівняння найбільш ефективних цифрових кореляційних приймачів ШПС (C_{xy},G_{xy}, F_{xy}) реалізовані у вигляді спецпроцесорів в унітарному ТЧБ.

Для аналізу ефективності застосування різних аналітичних виразів розрахунку кореляційних функцій одновимірних кодів Баркера розроблено програмне забезпечення алгоритмів вдосконаленої цифрової обробки нецентрованих кодів Баркера. Дана модель кодів Баркера необхідна для побудови програмного інструментарію цифрової обробки ШКП, які в каналах зв'язку піддаються впливу завад і на вхід АЦП цифрових кореляційних процесорів поступають у вигляді нецентрованих аналогових сигналів:

$$x(t)\to x_i+\delta(t),$$

де $x(t)$ - вхідний аналоговий сигнал на вході АЦП декодера; x_i - цифровий відлік решітчастої функції дискретизованого і квантованого

сигналу $x(t)$, який відповідає біту еталону коду Баркера; $\delta(t)$-аналогові значення адитивної завади в каналі зв'язку.

На виході АЦП формується потік цифрових даних у вигляді:

$$x_i^* = x_i \pm \Delta i,$$

де x_i^* - цифрове значення коду Баркера з врахуванням впливу завад; а $\pm \Delta i$ - цифрові значення завад.

В табл.13.2 приведені автокореляційні характеристики еталонного 11-бітного коду Баркера (табл. 13.3) з захисним інтервалом

1 1 1 -1 -1 -1 1 -1 -1 1 -1 0 0 0 0 0 0 0 0 0 0 0,

який формується у вигляді восьмирівневої нецентрованої кодової послідовності

8 8 8 0 0 0 8 0 0 8 0 4 4 4 4 4 4 4 4 4 4 4,

що обробляється цифровим кореляційним спецпроцесором у вигляді згортки решітчастих функцій (рис.13.6) згідно аналітичних виразів.

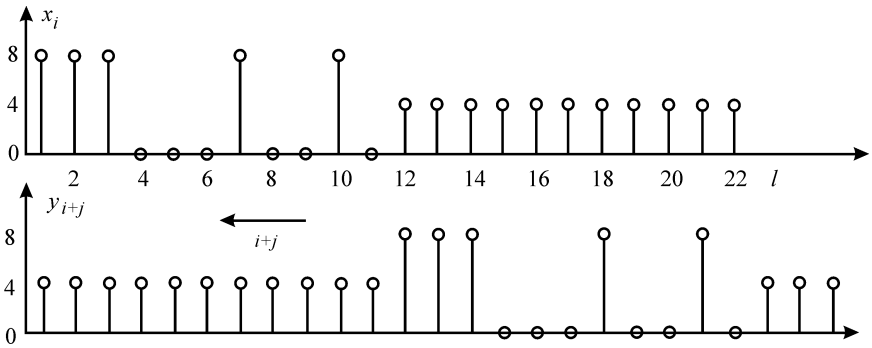
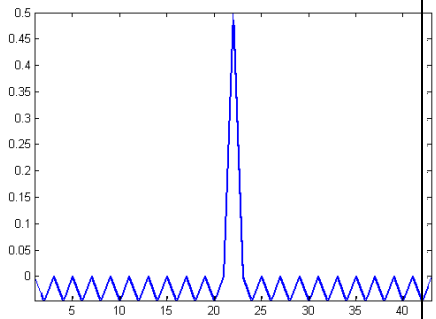
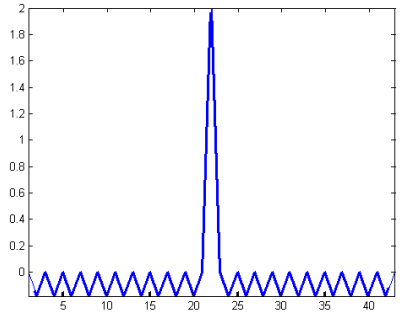


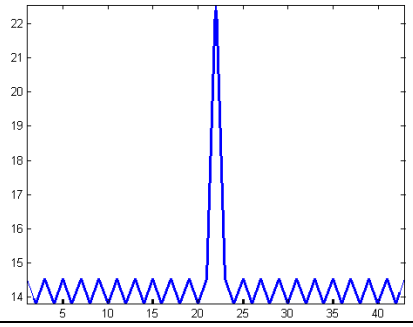
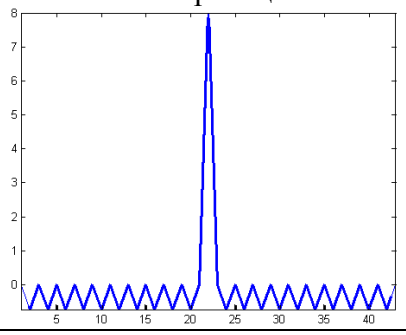
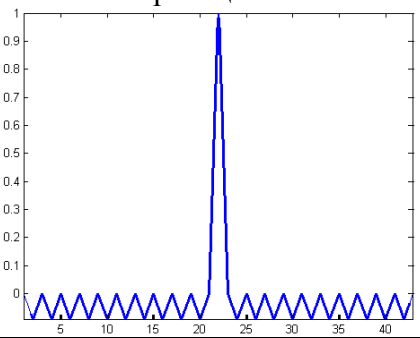
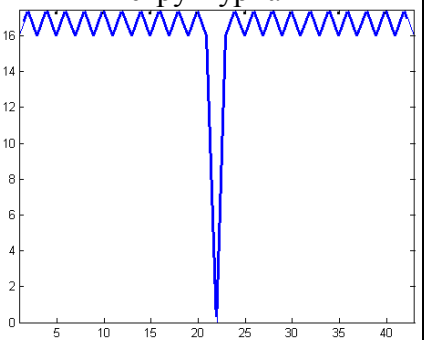
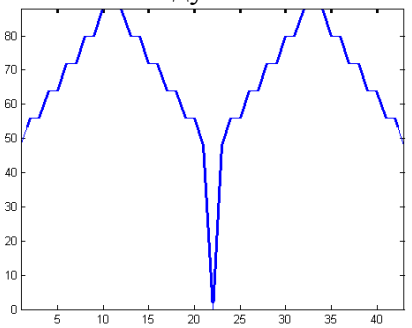
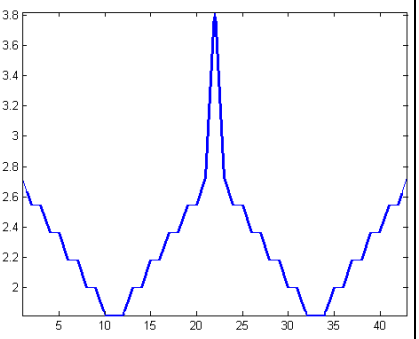
Рис.13.6. Решітчасті функції цифрової кореляційної згортки нецентрованих значень 11-бітного коду Баркера.

Таблиця 13.2.

Автокореляційні функції кодів Баркера

Кореляційна функція та її графік	
знакова	релейна
	

Продовження таблиці 13.2.

коваріаційна 	кореляційна 
нормована кореляційна 	структурна 
модульна 	еквівалентності 

В табл.13.1 видно, що спрощення алгоритмів цифрової згортки кодів Баркера приводить до відповідного погіршення характеристик кореляційних функцій. Очевидно, що причиною такого ефекту є невідповідність аналітики модульної $G_{xx}(j)$ та еквівалентної $F_{xx}(j)$ кореляційних функцій умові мультиплікативного перемноження кодів Баркера. Тому може бути справедлива гіпотеза, що для покращення кореляційних характеристик названих функції можливе існування інших ШКП.

Для передавання інформації використовують, як правило, кілька ШКП, наприклад, одну - для передавання нулів, а іншу - для передавання одиниць. Часто в якості другої ШКП використовують інверсію або фазовий зсув першої, при цьому головною умовою для вибору пари послідовностей є мінімальні значення бокових пелюсток кореляційних згорток на границях маніпульованих ШКП: прикладом генератора таких маніпульованих ШКП є структура, в якій використовується інверсія ключа коду Галуа. Перевагою кодів Баркера по відношенню до М-послідовностей максимальної довжини в даному випадку є відсутність перехідних процесів за рахунок нульових захисних інтервалів.

Одним з перспективних підходів до розвитку теорії кодів Баркера та вдосконалення їх кореляційних характеристик є пошук більш складних структур (двовимірності, багаторівневості і т.д.).

13.3. Метод спектрального аналізу сигналів рандомізованих в кодах поля Галуа.

При обробці сигналів в комп'ютерних та телекомунікаційних системах у багатьох випадках доводиться вимірювати спектри. Так, в задачах розпізнавання мови спектральний аналіз, як правило, передує подальшій спеціальній обробці. В системах стиснення смуги мовних сигналів [спектральний аналіз є основною операцією. Аналіз спектру частот проводиться при діагностуванні стану об'єкта, наприклад, бурової колони, глибинних насосів та іншого нафтового обладнання.

Класично для отримання спектру сигналу використовуються алгоритми дискретного і швидкого перетворення Фур'є. Для виконання цих алгоритмів необхідно проводити операції з комплексними числами, крім того має місце великий обсяг обчислень, що часто утруднює апаратну реалізацію методу. Тому актуальною задачею є пошук нових підходів у проведенні спектрального аналізу.

Метод спектрального аналізу, що базується на основі процедури функціональної рандомізації. Розглянемо його суть.

Рандомізацію сигналу доцільно виконувати з приведенням до форми М-послідовності, характерною особливістю яких є те, що їх автокореляційна функція:

$$R_{xx}(j) = \frac{1}{n}\sum_{i=1}^{n} \overset{o}{x}_i \cdot \overset{o}{x}_{i+j}, \qquad j = \overline{0,m};$$

при j=0 має пік, в інших точках її значення близьке до нуля. Це показано на рис. 13.7.

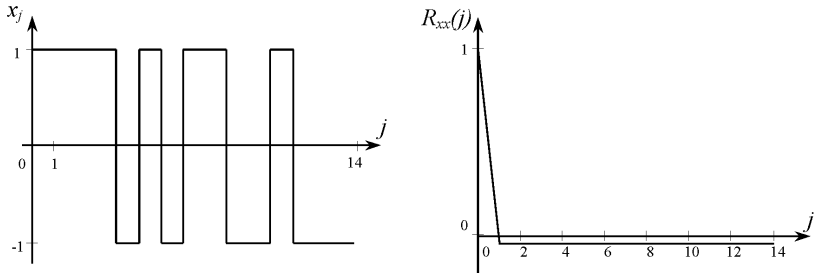


Рис.13.7. П’ятнадцятиелементна М-послідовність та її АКФ.

Для вибірки гармонічного сигналу $X = \{x_i\}$, $i = \overline{1,n}$ можна виконати рандомізацію згідно закону $j = k_i$, де $K = \{k_i\}$ – масив-ключ (закон) процедури рандомізації $\Re$an, який утворено таким чином, щоб огинаюча рандомізованого сигналу максимально наближалася до форми сигналу М-послідовності. Для підвищення точності рандомізації доцільно виконати вагову рандомізацію із введенням додаткового масиву вагових коефіцієнтів:

$$v_i = \begin{cases} 1 - x_i, & при \;\; 0 \le x_i \le 1, \\ -1 - x_i, & при \;\; -1 \le x_i \le 0. \end{cases}$$

Таким чином, шляхом виконання функціональної рандомізації можна виконати перетворення гармонічного сигналу у псевдовипадковий. Ця процедура наведена на рис.13.8. На рис.13.9 наведено графік автокореляційної функції рандомізованого сигналу без вагових коефіцієнтів (лінія 1) та з їх використанням (лінія 2).

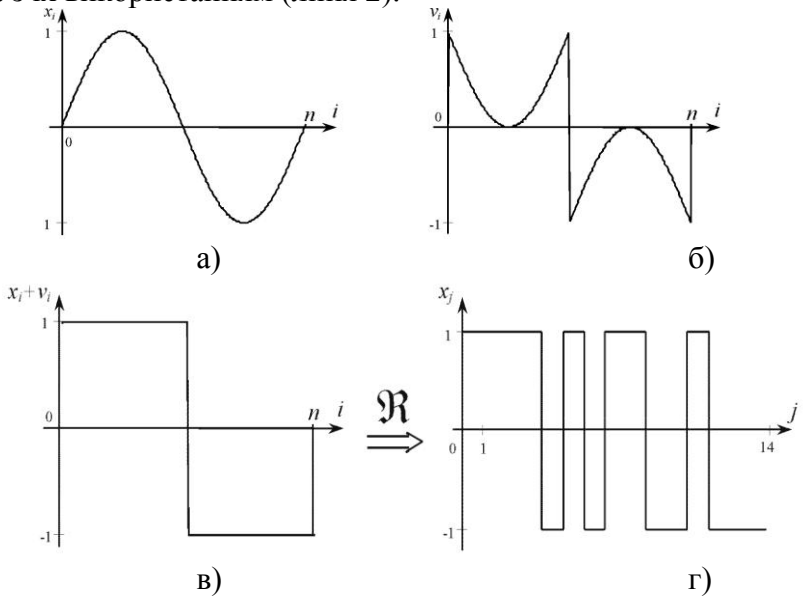


Рис.13.8. Процедура сигнальної рандомізації гармонічного сигналу.

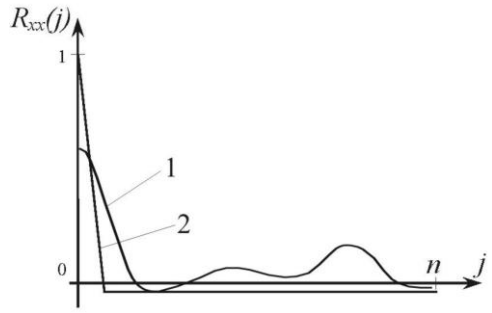


Рис.13.9. Автокореляційна функція рандомізованого сигналу.

Нехай необхідно перевірити вибірку гармонічного сигналу $X = \{x_i\}$, $i = \overline{1,n}$ на наявність деякої частоти ω_j, $j = \overline{1,m}$. Для цього необхідно мати матрицю B, рядками якої є вибірки гармонічних сигналів тестових частот, які є кратними по відношенню до основної частоти. Якщо об'єм кожної вибірки рівний n, то матриця буде мати розмір $m \times n$:

$$B = \begin{vmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{vmatrix}$$

- вибірка сигналу тестової частоти ω

- вибірка сигналу тестової частоти $2 \cdot \omega$

- вибірка сигналу тестової частоти $m \cdot \omega$

Необхідно також обчислити матрицю вагових коефіцієнтів V до елементів матриці B:

$$V = \begin{vmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{vmatrix}$$

Для виконання рандомізації потрібно генерувати матрицю K, елементами якого є закони рандомізації для кожної з тестових частот:

$$K = \begin{vmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mn} \end{vmatrix}$$

Елементи матриці $K=\{k_{ij}\}$, де $i=\overline{1,n}$, $j=\overline{1,m}$ утворюються таким чином, щоб огинаюча рандомізованого сигналу максимально наближалася до форми сигналу М-послідовності.

Таким чином, для дослідження вибірки сигналу $X=\{x_i\}$ необхідно мати матрицю вагових коефіцієнтів V та матрицю законів рандомізації K.

Перша операція, яка виконується при аналізі спектру сигналу полягає в додаванні елементів сигналу x_i із відповідними елементами рядків матриці вагових коефіцієнтів v_{ij}:

$$X^{V}=\begin{vmatrix} x_1+v_{11} & x_2+v_{12} & \dots & x_n+v_{1n} \\ x_1+v_{21} & x_2+v_{22} & \dots & x_n+v_{2n} \\ \vdots & \vdots & & \vdots \\ x_1+v_{k1} & x_2+v_{k2} & \dots & x_n+v_{kn} \end{vmatrix}$$

Далі виконується рандомізація отриманої матриці X^{V} відповідно до K:

$$X^{\Re an}=\Re an(X^{V})=\left\|x_{ij}^{\Re an}\right\|$$

Отримана матриця використовується для обчислення матриці коефіцієнтів кореляції:

$$\left\|R_{xx_{ij}}\right\|=\frac{1}{n}\sum_{m=1}^{n}x_{ij}^{\Re an}x_{i(j+h)}^{\Re an},\quad i=\overline{1,n},\ j=\overline{1,m},\ h=\overline{1,n}\ .$$

Для оцінки коефіцієнтів кореляції необхідно ввести пороговий коефіцієнт $\alpha=(0.7\div 0.8)A$, де A - значення піка кореляційної функції R_{xx}.

Останній крок методу – порівняння елементів матриці коефіцієнтів кореляції $R_{xx_{ij}}$ із пороговим коефіцієнтом α. За результатом порівняння робиться висновок про присутність в досліджуваному сигналі j-ої гармоніки сигналу: якщо в j-му рядку матриці R є елемент, значення якого перевищує значення коефіцієнта α, то в досліджуваному сигналі присутня j-та гармоніка.

13.4. Коди Баркера та М-послідовності у базисі Хаара-Галуа.

Сучасні спецпроцесори низових рівнів РКС оснащені вихідними модулями, які формують широкосмугові кодові послідовності, що класифікуються як ШКП. Тому при розробці та моделюванні досліджуваного класу СП доцільно виконати аналіз теоретичних основ їх побудови та системних характеристик.

В сучасних цифрових системах передавання інформації в якості ШКП використовуються кодові послідовності: Баркера; Лежандра; Цірлера; Пелі-Плоткіна; Френка; Галуа (М-сигнали); Голда, Касамі, Голея, коди побудовані на основі функцій Уолша, модифіковані М-сигнали та інші.

Найбільшого поширення та популярності набули коди Баркера, які при невеликій довжині володіють хорошими кореляційними властивостями (і практично, найкращими серед всіх інших одновимірних відомих ШКП) та дозволяють максимально ефективно використати канал зв'язку. Дані сигнали представляються двійковими компонентами $S_j = \pm 1$, в яких максимальний рівень бокових пелюсток не перевищує $1/n$ від основного піку, де n – кількість розрядів кодової послідовності.

Взаємокореляційну функцію інформаційних та еталонних кодів Баркера розраховують за формулою

$$\phi_{(x,y)}(j) = \sum_{i=1}^{n} sign(\overset{\circ}{x}_i) \cdot sign(\overset{\circ}{y}_{i+j}); \; j = 0,1,..., n,$$

де x_i – біт коду Баркера; y_i – тристабільний порівнюваний сигнал; $S_i = \pm 1,0$;

$$sign(\overset{\circ}{y}_{i+j}) = \begin{cases} 1, y_{i+j} > 0; \\ 0, y_{i+j} = 0; \\ -1, y_{i+j} < 0. \end{cases}$$

Усі відомі ШКП Баркера та їхні циклічні кореляційні функції подані в табл.13.3. Графіки кореляційних функцій Баркера зображено на рис.13.10, з якого видно, що із зростанням довжини ШКП знижується рівень бокових пелюсток.

Таблиця 13.3.

ШКП Баркера та їх кореляційні функції

№	k	Сигнали Баркера	Кореляційна функція
1	3	1,1,-1	-1,0,3
2	4	1,1,1,-1	-1,0,1,4
3	4	1,1,-1,1	1,0,-1,4
4	5	1,1,1,-1,1	1,0,1,0,5
5	7	1,1,1,-1,-1,1,-1	-1,0,-1,0,-1,0,7
6	11	1,1,1,-1,-1,-1,1,-1,-1,1,-1	-1,0,-1,0,-1,0,-1,0,-1,0,11
7	13	1,1,1,1,1,-1,-1,1,1,-1,1,-1,1	1,0,1,0,1,0,1,0,1,0,1,0,13

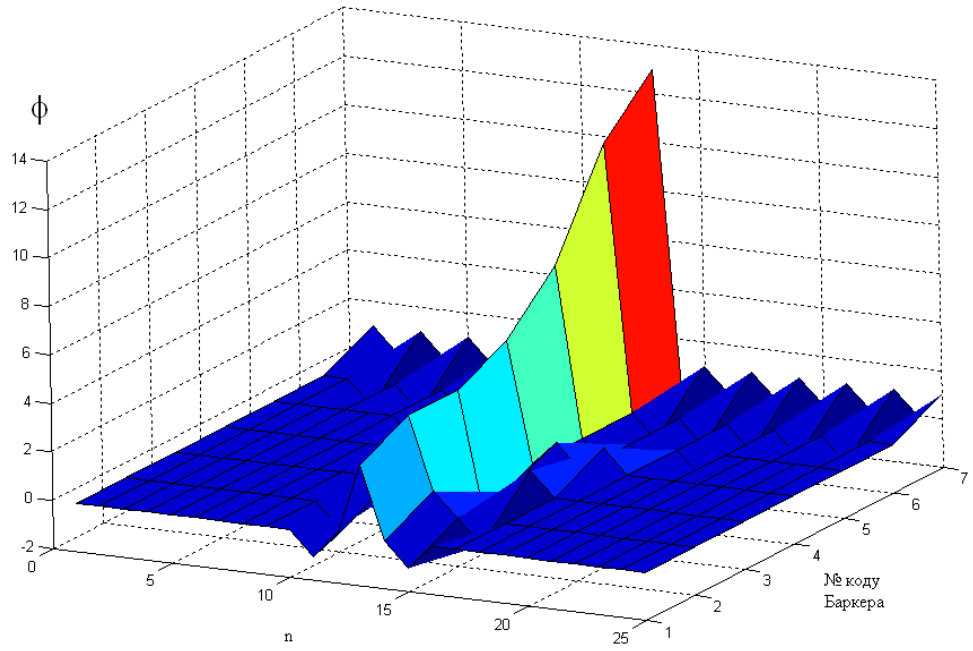


Рис.13.10. Кореляційні функції ШКП Баркера.

Незважаючи на хороші кореляційні властивості, коди Баркера мають один суттєвий недолік, а саме малу довжину $n \leq 13$, що обмежує можливості їх ефективного використання в каналах з високим рівнем завад та дозволяє створювати лише невелику кількість кодорозділених каналів зв'язку. А тому при необхідності отримання вищих параметрів системи (завадостійкості, дальності, прихованості, захищеності від несанкціонованого доступу) використовують інші види ШКП.

Сигнали на основі символів Лежандра являють собою клас сигналів з трійковими компонентами S_j=[1;0;-1]. Кількість компонентів визначається із наступного виразу $n=(p^n-1)/(p-1)$, де p-просте число, n-ціле число. Циклічні автокореляційні функції символів Лежандра мають бокові компоненти, що рівні нулю. Функції невизначеності при великих значеннях n мають бокові компоненти порядку $n^{(1/2)}$. Кореляційні функції при порівняно невеликих значеннях n мають бокові пелюстки, що значно менші за $n^{(1/2)}$. Слід зазначити, що сигнали Баркера з непарною кількістю компонентів входять в клас цих сигналів.

Сигнали Цірлера, компоненти яких описуються наступним аналітичним виразом $e^{(j\cdot(2\pi/p)\ i)\ c}$, де p - просте число, i - ціле число, що менше за p, на кожній позиції можуть мати одне з p значень. Кількість компонентів сигналу Цірлера визначається з виразу p^n-1. Циклічні кореляційні функції мають бокові пелюстки, що по модулю рівні одиниці. Формуються ці сигнали з допомогою порівняно нескладних алгоритмів.

Сигнали Пелі-Плоткіна − сигнали з двійковими компонентами S_j=±l і мають кількість компонентів, що визначається з виразу $n=p=(4k$-l), де k -ціле число. Циклічні кореляційні функції мають бокові компоненти, що по модулю рівні одиниці.

Для передавання інформації на великі відстані по каналах зв'язку з високим рівнем завад найефективніше застосовуються М–послідовності або послідовності максимальної довжини, що формуються на основі неприводимих алгебраїчних поліномів згідно виразу

$$X_{i+1} = (x_i \cdot a_i \oplus x_{i-1} \cdot a_{i-1} \oplus \ldots \oplus x_{i-n} \cdot a_{i-n}),$$

де $a_i \in 0{,}1$ – двійкові значення неприводимого алгебраїчного полінома, що формує код рекурентного ключа для М – послідовності.

В табл.13.4 наведені кількісні характеристики циклічних кореляційних функцій M – сигналів довжиною до 15 біт, які розраховані згідно виразу

$$S = 100\% \cdot \frac{L}{n}, \qquad (13.2)$$

де L, n – відповідно максимальний рівень бокової та головної пелюсток циклічних кореляційних функцій.

В табл.13.4. видно, що більшість М–послідовностей (М-сигналів) мають невисокі кореляційні характеристики, але велика кількість цих кодів дозволяє знайти компроміс між довжиною коду та його кореляційними властивостями.

Розвиток теорії кодів Баркера дозволив виявити систему модифікованих кодів Баркера, які мають кореляційні функції з боковими пелюстками -1, -2 та 1, 2, але за рахунок більшої довжини забезпечують краще співвідношення сигнал/шум.

Таблиця 13.4.

Кодові характеристики М – сигналів

Двійковий формат М – сигналу	Довжина М – сигналу, біт	Рівень пелюстки, S %
100	3	33.333
1110	4	25
11101	5	20
1110010	7	14.286
10110111000	11	9.091
1010110011111	13	7.692
10000001100101	14	14.286
111101011001000	15	13.333

Для виявлення ефективних кодових послідовностей даного класу сигналів використано залежність між довжиною М-сигналу та його десятковим значенням та проведено апроксимацію методом неперервних дробів, в результаті чого отримано аналітичний вираз для визначення області можливого існування М-сигналів з прийнятними кореляційними властивостями:

$$\mathrm{Re}\, g(l) = \frac{28960.94 - 22374.85 \cdot l + 3697.458 \cdot l^2 - 597.3114 \cdot l^3}{-6124.188 + 1036.755 \cdot l - 57.21558 \cdot l^2 + 1 \cdot l^3}, \qquad (13.3)$$

де l − довжина послідовності.

Графік можливого існування ефективних модифікованих кодів Баркера та М-послідовностей згідно виразу (13.3) зображено на рис.13.11.

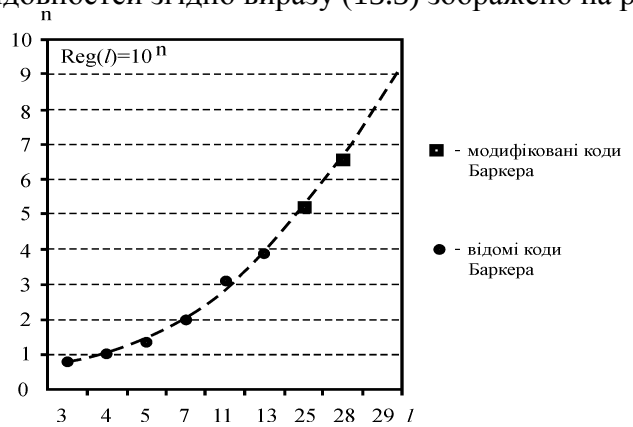


Рис.13.11. Залежність десяткового значення коду від довжини коду Баркера.

В результаті аналізу кореляційних характеристик модифікованих М-послідовностей згідно виразу (13.3) виявлені модифіковані коди Баркера з особливими кореляційними властивостями при $l \leq 29$ (табл.13.5).

Результати розрахунку характеристик модифікованих кодів Баркера при $l > 25$, розрахованих згідно виразу (13.2) представлений на рис.13.12, який демонструє системні переваги модифікованих кодів Баркера по відношенню до відомих кодів Баркера.

Отже, важливими системними властивостями одномірних ШКП є довжина кодової послідовності, співвідношення основного піку до рівня бокових пелюсток в автокореляційній функції та рівень взаємокореляції між ШПС, що використовуються в системі.

Таблиця 13.5.

Кількісні характеристики М-сигналів

Розрядність коду, l	Десятковий формат	Двійковий формат М-послідовності	Рівень пелюстки, S %
25	19218663	1001001010100000011100111	8.000
	23375843	1011001001010111111100011	8.000
	26208845	1100011111110101001001101	8.000
	30278985	1110011100000010101001001	8.000
28	150016301	1000111100010001000100101101	7.143
	153436391	1001001001010100000011100111	7.143
	187006748	1011001001010111111100011100	7.143
	208672178	1100011100000001010110110010	7.143
	228869240	1101101001000100010001111000	7.143
	236838308	1110000111011101110110100100	7.143
	242231881	1110011100000010101001001001	7.143
29	268848938	10000000001100100111100101010	10.345
	268879658	10000000001101100011100101010	10.345
	268901162	10000000001110001101100101010	10.345

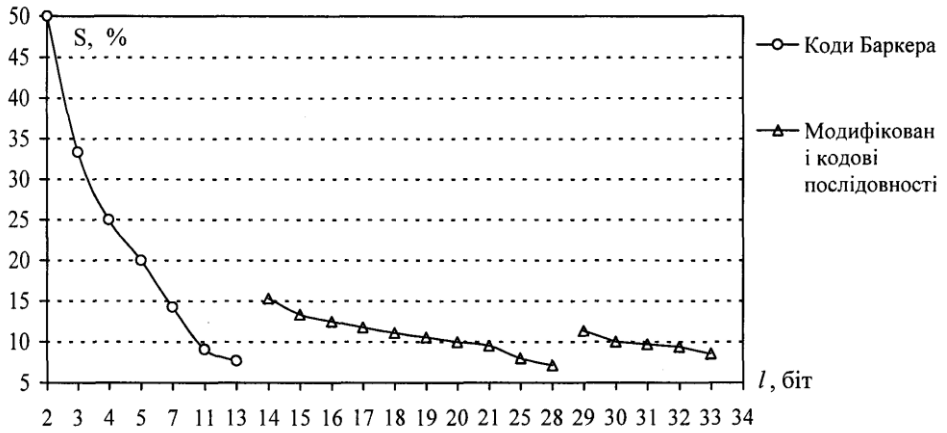


Рис.13.12. Залежність максимуму бокової пелюстки від довжини послідовності для кодів Баркера та модифікованих кодових послідовностей.

Обмежена кількість відомих шумоподібних послідовностей, їх невисокі кореляційні властивості при малій довжині коду та зростання популярності ШПС, системи вимагають пошуку нових ефективних ШКП. Перспективним є дослідження теоретичних засад та розвиток методів пошуку двовимірних ШКП, які мають кращі кореляційні властивості при заданій довжині кодової послідовності.

13.5. Теорія та системні характеристики двовимірних кодів Баркера.

Проаналізувавши відомі та модифіковані коди Баркера від 3 до 25 розрядів, можна зробити висновок, що ефективні ШКП мають приблизно рівну кількість нулів та одиниць у своєму коді, тобто

$$\sum_i a_i - \sum_j b_j = \pm 1;\ i+j=l, \tag{13.4}$$

де $\sum_i a_i$ - кількість нульових елементів, а $\sum_j b_j$ - кількість одиничних елементів.

Причому рівність у виразі (13.4) справедлива лише для ШКП, які мають максимальний рівень бокової пелюстки –1. Аналіз табл. 13.5 та кореляційних характеристик модифікованих кодів Баркера показує, що діапазон зміни максимальних значень бокових пелюсток автокореляційних функцій при зростанні розрядності кращих знайдених кодів в діапазоні розрядностей 25-32 біт не перевищує величину ± 4. Тому теоретичні границі модифікованих кодів Баркера великої розрядності можна представити у вигляді

$$\sum_i a_i - \sum_j b_j \le \pm 4;\ \ i + j = l\,. \tag{13.5}$$

На рис.13.13 показана кореляційна характеристика модифікованого коду Баркера -1 1 1 -1 1 1 -1 1 -1 1 -1 1 1 1 1 1 1 -1 -1 -1 1 1 -1 -1 -1 (*l=25*), який відповідає умові (13.5).

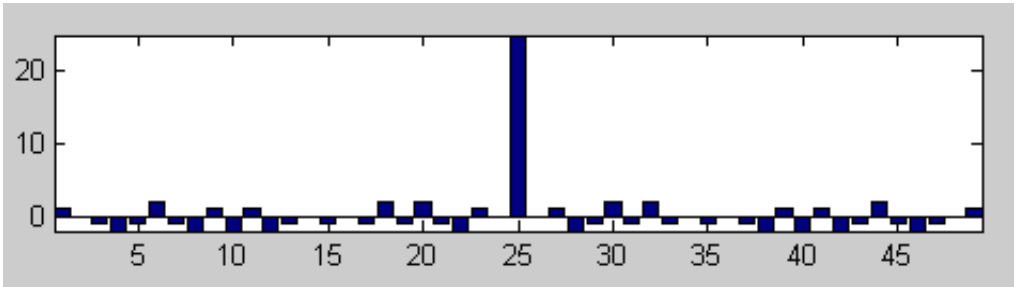


Рис.13.13. Кореляційна функція модифікованого коду Баркера.

Встановлена властивість відомих одновимірних ШКП дозволяє суттєво звузити границі їх пошуку внаслідок лінійної характеристики зони їх існування при представленні числових значень кодових послідовностей в залежності від розрядності у двійковому логарифмічному масштабі, що показано на рис.13.14.

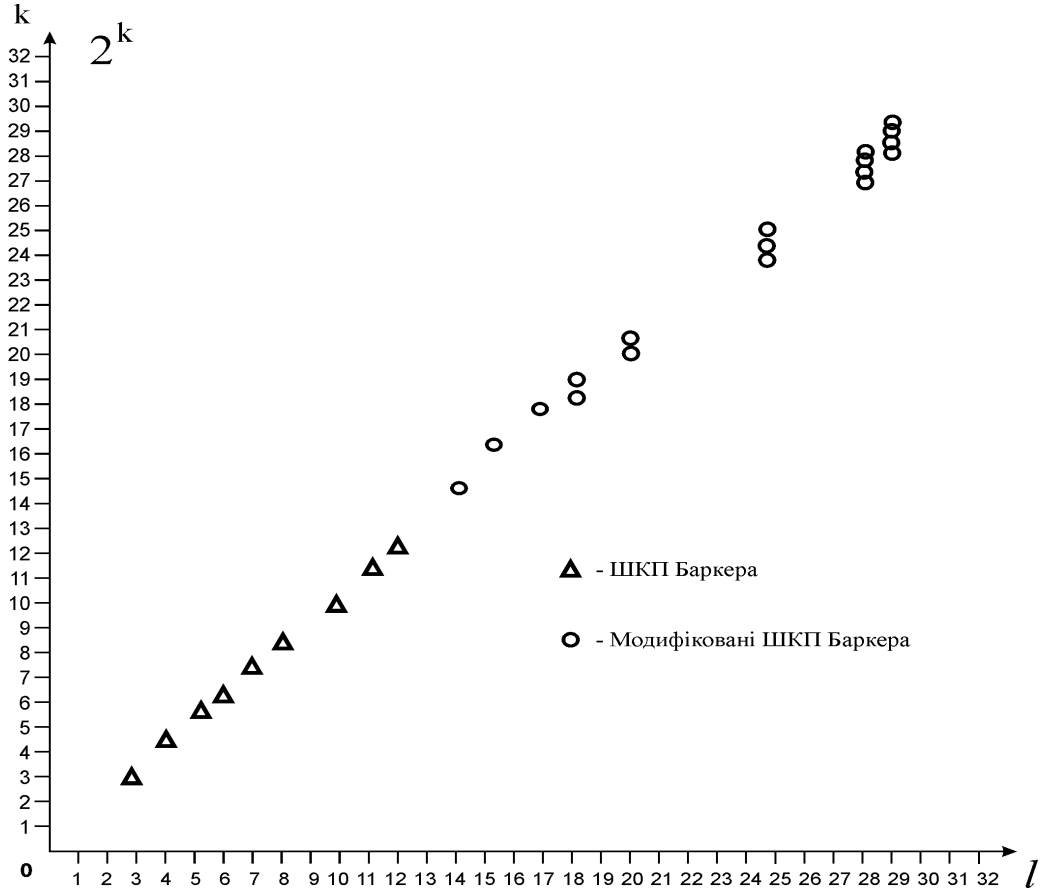


Рис.13.14. Позиції ШКП та їх числові представлення у двійковій системі числення.

З даного рисунку видно, що зі зростанням довжини кодової послідовності збільшується її діапазон представлення, а, отже, і кількість кодів для даної розрядності.

Використання даної залежності дозволяє значно скоротити час пошуку ефективних ШКП великої розрядності. Для пошуку двовимірних ШКП запропоновано рівняння (13.5) у вигляді:

$$\sum_i a_i - \sum_j b_j \le \pm 4; \quad i + j = h \cdot m, \tag{13.6}$$

де *h, m* – відповідно число рядків та стовпчиків матриці двовимірного коду.

Розроблена аналітика розрахунку кореляційних характеристик ШКП на основі двовимірних кореляційних функцій, які враховують різні можливі способи перемноження матриць двовимірних кодів:

$$K_{(x,y)} = \sum_{j=1}^{h}\sum_{i=1}^{m} x_{i,j} \cdot y_{i,j} + \sum_{i=1}^{m}\sum_{j=1}^{h} x_{i,j} \cdot y_{i,j};$$

$$K_{(x,y)} = \sum_{j=1}^{h}\sum_{i=1}^{m} x_{i,j} \cdot y_{i,j} + \sum_{i=1}^{m}\sum_{j=h}^{1} x_{i,j} \cdot y_{i,j};$$

$$K_{(x,y)} = \sum_{j=1}^{h}\sum_{i=1}^{m} x_{i,j} \cdot y_{i,j} + \sum_{i=m}^{1}\sum_{j=1}^{h} x_{i,j} \cdot y_{i,j}; \quad K_{(x,y)} = \sum_{j=1}^{h}\sum_{i=1}^{m} x_{i,j} \cdot y_{i,j} + \sum_{i=m}^{1}\sum_{j=h}^{1} x_{i,j} \cdot y_{i,j},$$

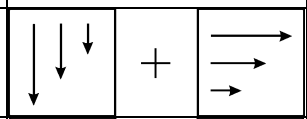
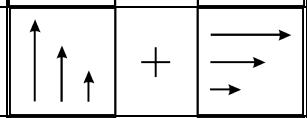
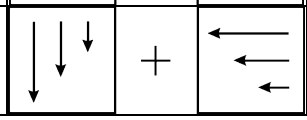
де x_{ij}, y_{ij} – відповідно елементи інформаційного та еталонного двовимірних кодів Баркера.

При цьому в процесі двовимірної кореляційної згортки ДШКП виконується попередня рандомізація елементів матриці еталонного коду.

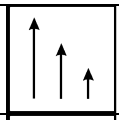
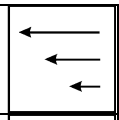
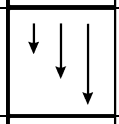
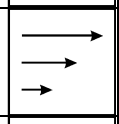
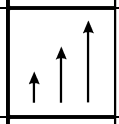
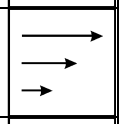
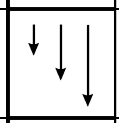
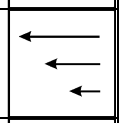
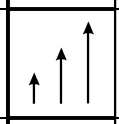
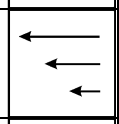
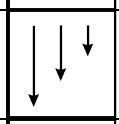
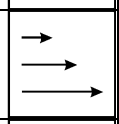
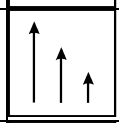
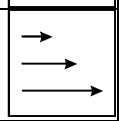
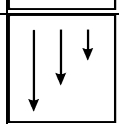
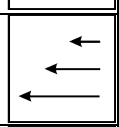
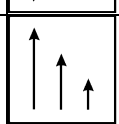
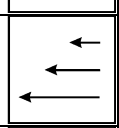
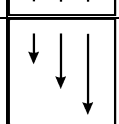
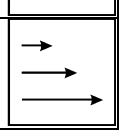
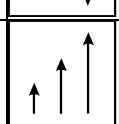
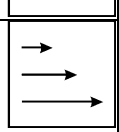
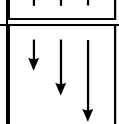
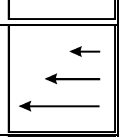
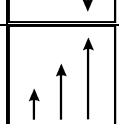
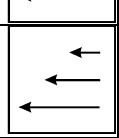
В табл.13.6 представлені різні алгоритми цифрової кореляційної обробки двовимірних ШКП.

Таблиця 13.6.

Алгоритми кореляційної обробки двовимірних ШКП

№	Графічне представлення	Вираз
1	2	3
1		$K(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{h} F(x_{i,j}, y_{i,j}) + \sum_{j=1}^{h}\sum_{i=1}^{m} F(x_{i,j}, y_{i,j})$
2		$K(x,y) = \sum_{i=1}^{m}\sum_{j=h}^{1} F(x_{i,j}, y_{i,j}) + \sum_{j=1}^{h}\sum_{i=1}^{m} F(x_{i,j}, y_{i,j})$
3		$K(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{h} F(x_{i,j}, y_{i,j}) + \sum_{j=1}^{h}\sum_{i=m}^{1} F(x_{i,j}, y_{i,j})$

Продовження таблиці 13.6.

4		+		$K(x,y)=\sum_{i=1}^{m}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=1}^{h}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
5		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=1}^{h}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
6		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=1}^{h}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
7		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=1}^{h}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
8		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=1}^{h}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
9		+		$K(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
10		+		$K(x,y)=\sum_{i=1}^{m}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
11		+		$K(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
12		+		$K(x,y)=\sum_{i=1}^{m}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
13		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
14		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=1}^{m}F(x_{i,j},y_{i,j})$
15		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=1}^{h}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$
16		+		$K(x,y)=\sum_{i=m}^{1}\sum_{j=h}^{1}F(x_{i,j},y_{i,j})+\sum_{j=h}^{1}\sum_{i=m}^{1}F(x_{i,j},y_{i,j})$

В таблиці 13.6 в якості функцій цифрової кореляції $K(x, y)$ можуть бути ефективно використані аналітичні вирази з табл.13.1 $H_{xy}(j)$, $B_{xy}(j)$, $K_{xy}(j)$, $R_{xy}(j)$, $\rho_{xy}(j)$, $C_{xy}(j)$, $G_{xy}(j)$, $F_{xy}(j)$, а в якості функцій $F(x_{i,j}, y_{i,j})$ відповідно: $sign(\dot{x}_i) \cdot sign(\dot{y}_{i+j})$ – для знакової; $\dot{x}_i \cdot sign(\dot{y}_{i+j})$ – для релейної; $x_i \cdot y_{i+j}$ – коваріаційної; $(\dot{x}_i \cdot \dot{y}_{i+j})/\delta_x \cdot \delta_y$ – нормалізованої кореляційної; $(x_i \cdot y_{i+j})^2$ – структурної; $|x_i| \cdot |y_{i+j}|$ – модульної; $\overset{\vee}{Z}(x_i, y_{i+j})$ – еквівалентності.

При цьому для аналізу впливу завад в каналах зв'язку на двовимірні ШКП, може бути використаний розроблений метод цифрової кореляційної обробки кодів Баркера на основі їх багаторівневих центрованих моделей.

Розроблені теоретичні області існування одновимірних ШКП та можливої області існування двовимірних ШКП, а також аналітика кореляційних характеристик двовимірних ШКП лягла в основу програмного інструментарію моделювання та їх пошуку при заданих характеристиках розрядності, а також максимальних значень основних та бокових пелюсток кореляційних функцій.

13.6. Моделювання автокореляційних характеристик двовимірних кодів Баркера.

Комп'ютерне моделювання характеристик двовимірних ШКП проведене згідно виразів (13.6), (13.7) та аналітичних виразів з табл.13.1, при умові h=m, дозволило вперше отримати кодові матриці (рис.13.15) двовимірних ШКП, які задовільняють умові (13.6), та графіки їх кореляційних характеристик, що показані на рис.13.16.

1	1			1	1	0		1	1	1	1		1	0	0	1	0		
1	0			1	0	1		1	0	1	0		0	1	0	1	0		
				0	0	1		1	1	0	0		1	0	0	0	0		
								1	0	0	1		0	0	1	1	1		
													0	0	1	1	1		
а)				б)				в)					г)						

Рис.13.15. Матриці двовимірних кодів (а) 2х2, б) 3х3, в) 4х4, г) 5х5).

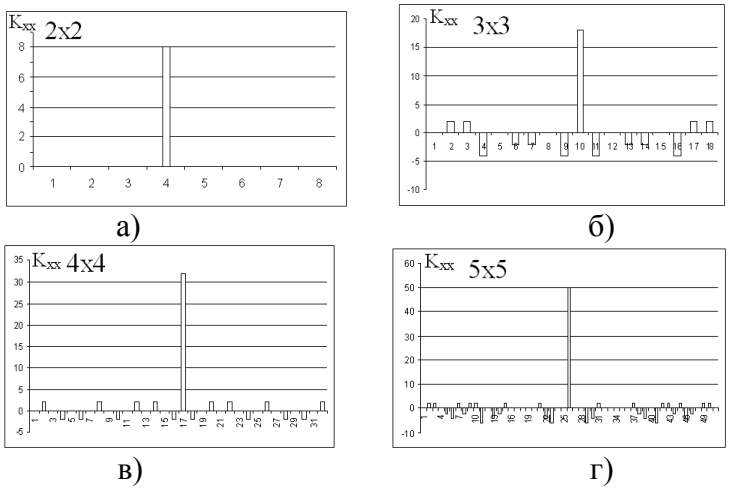


Рис.13.16. Графіки кореляційних властивостей двовимірних кодів для матриць.

На рис.5.17 показано область існування знайдених та прогнозованих двовимірних ШКП.

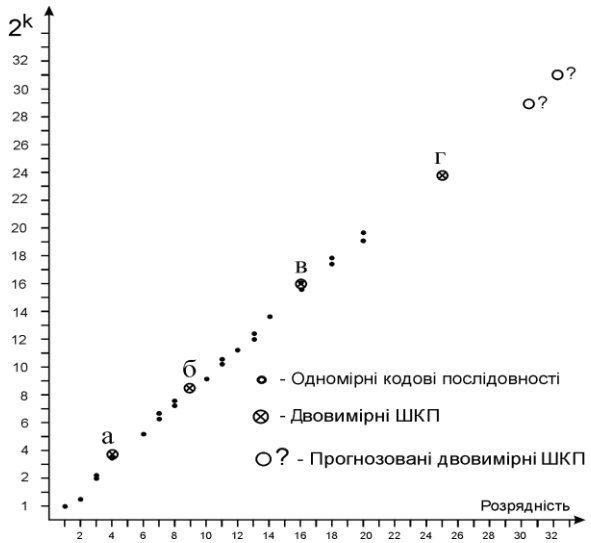


Рис.13.17. Області існування відомих одновимірних та знайдених двовимірних ШКП з найкращими кореляційними властивостями.

В результаті виконаного моделювання та пошуку двовимірних ШКП згідно умови (13.6) отримані їх характеристики для різних значень h, m, які розраховані згідно аналітичних виразів запропонованих алгоритмів двовимірної кореляційної обробки (табл.13.6) при виконанні умови (13.6). Кращі з отриманих кодових послідовностей розмірністю 3×3, 4×4, 5×5 представлені в табл.13.7.

Таблиця 13.7.

Знайдені двовимірні ШКП та їх кореляційні функції

Розмір, матриці та № алгоритму (табл.5.13)	Двовимірний ШКП + =1; - = -1	Бокова, пелюстка L_i max	Кореляційна функція	Головна пелюстка, L_0
1	2	3	4	5
3x3, (6)	---++--+-	2	18, 0, -6, 0, 2, 0, 2, 0, 2	18
	--+++-++-		18, 0, -6, 0, 2, 0, -6, 0, 2	
	-+--++---		18, 0, -6, 0, 2, 0, 2, 0, 2	
	-++-+++--		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+--+---++		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+-++--+++		18, 0, -6, 0, 2, 0, 2, 0, 2	
	++---+--+		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+++--++-+		18, 0, -6, 0, 2, 0, 2, 0, 2	
3×3 (5)	-+-++----	2	14 -4 2 0 -4 2 2 0 2	14
	-+-+++---		14 -4 0 -6 0 -2 2 0 2	
	+-+---+++		14 -4 0 -6 0 -2 2 0 2	10
	+-+--++++		14 -4 2 0 -4 2 2 0 2	
3x3, (15)	---++--+-	2	14 -4 -2 0 0 2 2 0 2	14
	-+-+++---		14 -4 0 -6 0 -2 2 0 2	
	+-+---+++		14 -4 0 -6 0 -2 2 0 2	10
4x4, (5)	--++++-+--+---+-	0	32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	20
	-+---+--+-++++--		32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	
4x4, (6)	-+-+-++--+++----	2	32, -2, 0, -10, -4, -2, 4, 2, 0, -2, -4, 2, -4, 2, 0, 2	20

продовження таблиці 13.7

4x4, (7)	-+-+-++-- ++++---		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	20
	-++----+++-+- +--		32, -2, 0, -6, -4, -2, -4, -2, 0, 2, 4, 2, -4, -2, 0, 2	
4x4, (13)	-+---+--+- ++++--		32, -2, 0, -6, -4, -2, -4, -2, 0, 2, 4, 2, -4, -2, 0, 2	32
	-+-+-++--+++- ---		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
	-+-+-++-- ++++---		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	
	-++----+++-+- +--		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	
4x4, (13)	-++--+-+- ++++---		32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	32
	+--++++---+- +-++		32, -2, 0, -10, -4, -2, 4, 2, 0, -2, -4, 2, -4, 2, 0, 2	
	+-+-+--++--- ++++		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
5x5, (13)	-+---++---+-+- -+--++++++-	4	50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	50
	-++++++--+-- +-+---++---+-		50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	
	+------++-++- +-+++--+++-+		50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	

продовження таблиці 13.7

	+-+++--+++- +-++-++------+		50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2,	

Результати комп'ютерного моделювання двовимірних ШКП показали, що найвищий рівень головної пелюстки циклічної кореляційної функції забезпечують 1-й та 6-й алгоритми (табл.13.6). Найефективніші двовимірні коди з максимальною різницею між величиною головної та бокової пелюсток кореляційних функцій подані в табл.13.8.

Таблиця 13.8.

Найефективніші двовимірні ШКП та їх кореляційні функції

Розмір матриці	Двовимірний ШКП, - =1; + = -1	Бокова пелюстка L_i, max	Кореляційна функція	Головна пелюстка, L_0
3x3 (6)	---++--+-	2	18, 0, -6, 0, 2, 0, 2, 0, 2	18
	--+++-++-		18, 0, -6, 0, 2, 0, -6, 0, 2	
	-+--++---		18, 0, -6, 0, 2, 0, 2, 0, 2	
	-++-+++--		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+--+---++		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+-++--+++		18, 0, -6, 0, 2, 0, 2, 0, 2	
	++---+--+		18, 0, -6, 0, 2, 0, -6, 0, 2	
	+++--++-+		18, 0, -6, 0, 2, 0, 2, 0, 2	
4x4 (6)	----+++--++-+-+-	4	32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	32
	---+-+--+++-++--		32, -2, -4, -2, 4, 2, -8, 2, -4, -2, 0, -2, -4, 2, 4, 2	
	---++++--++-+-+-		32, -2, 0, -10, -4, -2, 4, 2, 0, -2, -4, 2, -4, 2, 0, 2	
4x4 (6)	---++++-+-+--++-	4	32, -2, 0, -6, -4, -2, -4, -2, 0, 2, 4, 2, -4, -2, 0, 2	32
	--+-+-+++----++-		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
	--++++-+--+---+-		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	
	--++++-+--+---+-		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	

продовження таблиці 13.8

	-+---+--+- ++++--		32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	
	-+-+-++--+++- ---		32, -2, 0, -10, -4, -2, 4, 2, 0, -2, -4, 2, -4, 2, 0, 2	
	-+-+-++-- ++++---		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
	-++----+++-+- +--		32, -2, 0, -6, -4, -2, -4, -2, 0, 2, 4, 2, -4, -2, 0, 2	
	-++--+-+- ++++---		32, -2, 0, -10, -4, -2, 4, 2, 0, -2, -4, 2, -4, 2, 0, 2	
	+--++++---+- +-++		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
	+-+-+--++--- ++++		32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	
	+-+++-++-+--- -++		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	
	++----+-++- +++-+		32, -2, 0, 2, -4, -2, -4, 2, -4, -2, -4, 2, 4, -2, 0, 2	
	++--+---++-+- +++		32, -2, -4, -2, 4, 2, -8, 2, -4, -2, 0, -2, -4, 2, 4, 2	
	++-+-+--- ++++--+		32, -2, 0, -10, -4, -2, 4, -2, -4, 2, 4, 2, 4, -6, 0, 2	
	+++-+-++---+- -++		32, -2, -4, -2, 4, 2, -8, 2, -4, -2, 0, -2, -4, 2, 4, 2	
	++++---++--+- +-+		32, -2, 0, -10, 0, 2, 0, -2, 4, -6, 0, -2, 0, 2, 0, 2	
5x5 (6)	-+---++---+-+-- +--++++++-	4	50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	50
	-++++++--+-- +-+---++---+-		50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	
	+------++-++- +-+++--+++-+		50, 0, -2, 0, 2, 4, -6, 0, -2, 4, -2, 0, -10, 4, 2, -4, -2, 4, -6, -8, -2, 0, 2, -4, 2	
	+-+++--+++-+- ++-++------+		50,0,-2,0,2,4,-6,0,-2,4,-2,0,-10,4, 2,-4,-2, 4, -6, -8, -2, 0, 2, -4, 2	

Результати комп'ютерного моделювання дозволили встановити, що існує значна кількість двовимірних ШКП з заданими кореляційними характеристиками, в яких максимальний рівень бокової пелюстки не перевищує рівень +2. Так, кількість знайдених кодів, які відповідають вказаній умові, а величина головної пелюстки кореляційної функції наближається до максимальної, становить: 3×3 – 352; коди; 4×4 – 758; 5×5 – 1448.

Аналіз ефективності знайдених двовимірних ШКП у порівнянні з одновимірними виконаємо згідно виразу (13.2), де L_0=h·m. Результати порівняння кореляційних характеристик двовимірних ШКП з одновимірними показані на рис.13.18.

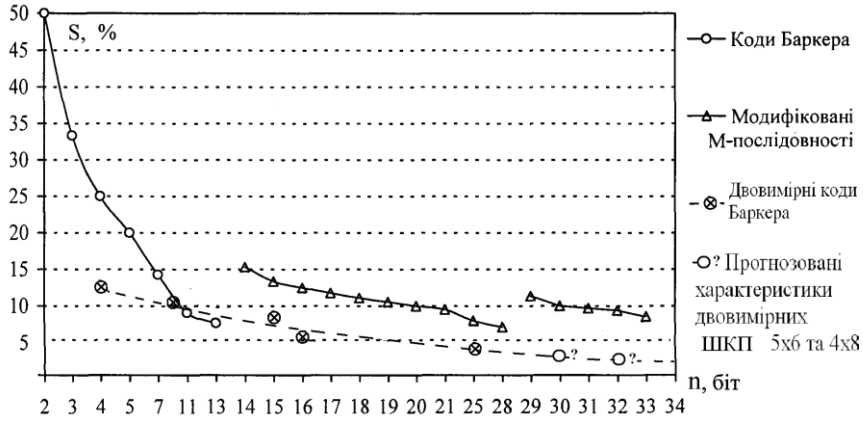


Рис.13.18. Залежність максимуму бокової пелюстки для кодів Баркера та інших ШКП від довжини коду.

З рис.13.18 видно, що знайдені двовимірні ШКП мають суттєво кращі характеристики ефективності згідно оцінки (13.2) по відношенню до відомих одновимірних та модифікованих кодів Баркера.

Оскільки найважливішою характеристикою завадозахищеності кодів Баркера при заданій швидкості передавання даних є відношення різниці між величиною головного пелюстка та максимальним значенням позитивного викиду бокового пелюстка кореляційної функції до розрядності коду, дослідження характеристик двовимірних ШКП доцільно виконати згідно аналітичного виразу:

$$V = \frac{(L_0 - L_{i\max})^2}{2n};\ L_0 - L_{i\max} > 0, \qquad (13.8)$$

де L_0, $L_{i\max}$ - відповідно значення головної та максимальної бокової пелюстки кореляційної функції, 2n враховує довжину ШКП Баркера з

захисною послідовністю нульових бітів, а коефіцієнт степені в чисельнику враховує ступінь завадозахищеності ШКП.

На рис.13.19 показано порівняння згідно критерію (13.8) характеристик відомих одновимірних та двовимірних ШКП в залежності від розрядності кодів n.

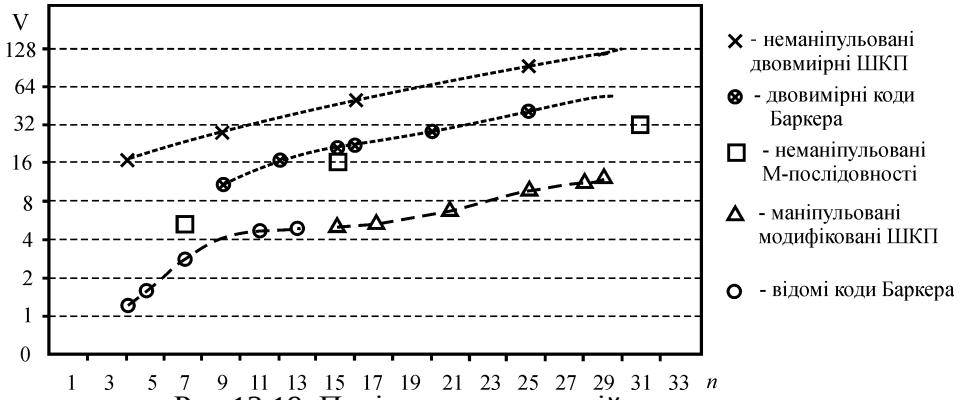


Рис.13.19. Порівняння кореляційних характеристик завадозахищеності одновимірних та двовимірних ШКП.

З рис.13.19 видно, що знайдені на основі розроблених теоретичних положень двовимірні коди Баркера та ШКП характеризуються стійкими перевагами, які зростають при зростанні розрядності кодових послідовностей.

При цьому найкращі кореляційні властивості, а відповідно максимальну завадозахищеність, мають знайдені неманіпульовані двовимірні ШКП (✕). Даний тип сигналів найефективніше можна використовувати в автономних сенсорах низових рівнів безпровідних РКС, які виконують функції охоронної сигналізації, формувачів сигналів синхронізації, ідентифікації відхилень технологічних параметрів від норми, екологічного моніторингу навколишнього середовища та інш.

Знайдені двовимірні коди Баркера (⊗) в класі досліджуваних кодів забезпечують максимальну швидкість передавання даних при заданій завадозахищеності. Дані кодові послідовності найефективніше застосувати в якості вихідних кодів розробленого класу спецпроцесорів низових рівнів РКС.

Кореляційні характеристики неманіпульованих одновимірних М-сигналів визначають верхню границю ефективності одновимірних ШКП і можуть використовуватися в аналогічних випадках застосування двовимірних неманіпульованих ШКП. Маніпуляцію даних кодів, як показано в роботі, найефективніше виконувати на основі інвертування

кодового ключа коду Галуа, що забезпечує мінімальні викиди бокових пелюсток кореляційних функцій на границях маніпульованих кодів.

Характеристики відомих кодів Баркера (○) та модифікованих М-послідовностей (△), представлені на рис.13.19, відображають верхню границю завадозахищеності одновимірних маніпульованих ШКП. Дані коди знайшли широке застосування в сучасних телекомунікаційних та комп'ютерних мережах при побудові спецпроцесорів низових рівнів РКС а також в стандартних технологіях DSSS та FHSS.

Аналіз графіків на рис.5.21 дозволяє оцінити переваги завадозахищеності розроблених двовимірних ШКП по відношенні до відомих кодів Баркера та одновимірних М-послідовностей згідно виразу:

$$S_{ШКП} = \frac{V_2}{V_1},$$

де V_2, V_1 – відповідно параметр ефективності двовимірних та одновимірних ШКП.

Порівняння виконаємо в діапазоні розрядностей знайдених двовимірних ШКП n=9-25, що відповідає зростанню завадозахищеності розроблених кодів при однаковій швидкості передавання даних в 3,03-3,35 рази.

Порівняння характеристик неманіпульованих одновимірних та двовимірних ШКП на основі екстраполяційного прогнозування в діапазоні n=7-31 відповідає відносному підвищенню завадозахищеності останніх в середньому в 2,6 рази.

Проведені дослідження характеризують потенційні можливості одновимірних та двовимірних ШКП без впливу завад в каналах зв'язку, які можуть приводити до спотворення кодових послідовностей та погіршення їх кореляційних властивостей. Тому доцільно провести дослідження рівня стійкості досліджуваного класу кодів шляхом моделювання впливу адитивних завад.

13.7. Теорія впливу помилок на двовимірні коди Баркера та їх симетрія.

Аналіз впливу завад на кореляційні характеристики двовимірних та одновимірних ШКП виконаємо шляхом порівняння погіршення їх характеристик згідно критерію (13.8) при появі в кодах однократних та багатократних помилок. В якості базових кодів ШКП з невеликою найбільш близькою розрядністю використаємо запропонований двовимірний ШКП Баркера розмірністю 4×4 та відомий одновимірний код Баркера довжиною 13 біт:

$$\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{matrix}, \qquad 1,1,1,1,1,-1,-1,1,1,-1,1,-1,1.$$

Оскільки обмін даними між віддаленими процесорами в комп'ютерних системах виконується у вигляді біт-орієнтованих кодів, двовимірні ШКП розгортаються і передаються у вигляді одновимірних кодових послідовностей, даний код приймає наступний вигляд:

$$1\,1\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,0\,1. \qquad (13.9)$$

В зв'язку з тим, що цифрова кореляційна обробка ШКП виконується на основі знакової функції $H_{xx}(j)$ (табл.13.1) над центрованими значеннями, біт-орієнтована кодова послідовність двовимірного ШКП переводиться з логічного базису у базис дискретних ортогональних функцій шляхом заміни символів "1" на "-1", а відповідно "0" на "1", тоді одновимірне логічне представлення двовимірного ШКП (13.9) отримає наступний вигляд:

-1 -1 1 1 1 1 -1 1 -1 -1 1 -1 -1 -1 1 -1

або, як це показано в табл.13.2, у вигляді послідовності символів

$$--++++-+--+----+-.$$

З врахуванням захисних нульових інтервалів кодових послідовностей Баркера дані базові коди для порівняння їх характеристик отримають вигляд:

1,1,1,1,1,-1,-1,1,1,-1,1,-1,1,0,0,0,0,0,0,0,0,0,0,0,0,0 – код Баркера;

-1,-1,1,1,1,1,1,-1,1,-1,-1,1,-1,-1,-1,1,1,-1, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 – двовимірний ШКП.

Для виявлення погіршення взаємокореляційних властивостей вибраних ШКП в досліджувану кодову послідовність, для якої розраховувалась взаємокореляційна функція, вводились однократні та багатократні помилки, які полягали в інвертуванні одного чи кількох бітів коду. Оскільки досліджувані коди при інвертуванні певних бітів мало чутливі до цих змін, то з отриманих результатів вибирались ті, що приводять до найбільшого зниження ефективності кодової послідовності. Кодові послідовності "змінених" ШКП та їх кореляційні функції наведені у табл.13.9 з врахуванням однократних та багатократних помилок. З метою оцінки нижніх границь погіршення властивостей досліджуваних кодів в табл.13.9 та 13.10 приведені кореляційні характеристики та графіки при максимальному впливі відповідного числа пошкоджених бітів ШКП. При відборі кодів з максимально погіршеними характеристиками застосований генетичний алгоритм, який в кожній наступній ітерації при зростанні числа

пошкоджених бітів вибирає варіант з максимальним впливом попередньої ітерації.

Таблиця 13.9.

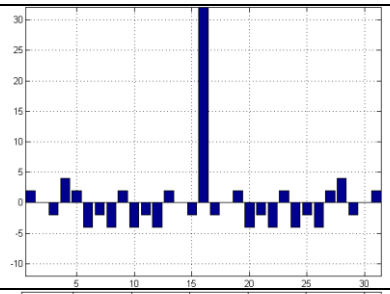
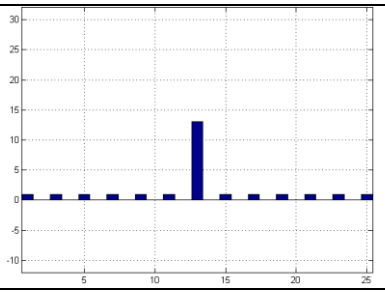
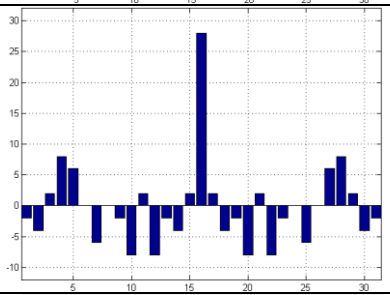
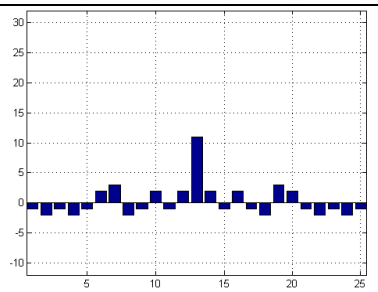
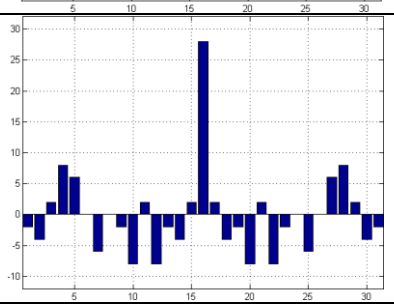
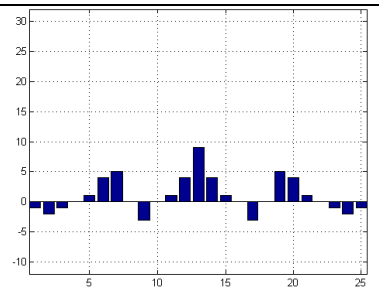
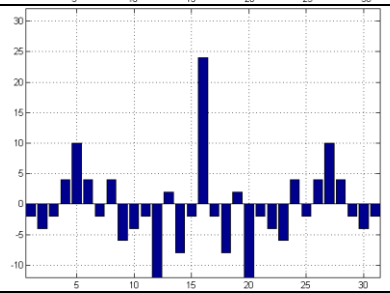
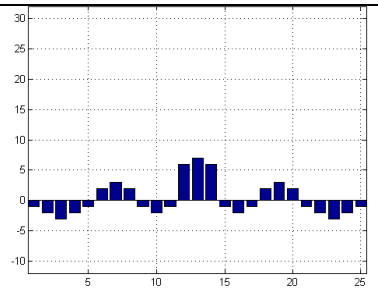
Кореляційні функції ШКП з помилками.

К-ть помилок	Двовимірний ШКП	Одновимірний код Баркера
0	-1 -1 1 1 1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 -1 32,-2, 0, 2,-4,-2,-4, 2,-4,-2,-4, 2, 4,-2, 0, 2	1 1 1 1 1 -1 -1 1 1 -1 1 -1 1 13, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1
1	-1 -1 1 1 1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 28,2,-4,-2,-8,2,-8,-2,0,-6,0,6,8,2,-4,-2	1 1 1 1 1 -1 -1 1 1 -1 1 -1 -1 11,2,-1,2,-1,-2,3,2,-1,-2,-1,-2,-1
2	-1 -1 1 1 1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 28,2,-4,-2,-8,2,-8,-2,0,-6,0,6,8,2,-4,-2	1 1 1 1 1 -1 -1 1 1 1 1 -1 -1 9,4,1,0,-3,0,5,4,1,0,-1,-2,-1
3	-1 -1 1 1 1 1 -1 1 -1 -1 1 -1 -1 1 1 1 24,-2,-8,2,-12,-2,-4,-6,4,-2,4,10,4,-2,-4,-2	1 1 1 1 1 -1 -1 1 1 1 -1 -1 -1 7,6,-1,-2,-1,2,3,2,-1,-2,-3,-2,-1
4	-1 -1 1 1 1 1 -1 1 -1 -1 -1 -1 -1 1 1 1 20,2,-4,-2,-8,-6,-8,-10,0,2,8,10,4,-2,-4,-2	1 1 1 1 1 -1 -1 -1 1 1 -1 -1 -1 5,8,1,-4,-3,0,1,0,-1,-2,-3,-2,-1
5	-1 -1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 16,6,-8,-6,-12,-10,-4,-6,0,2,8,10,4,-2,-4,-2	1 1 1 1 1 -1 -1 -1 1 1 -1 -1 -1 5,8,1,-4,-3,0,1,0,-1,-2,-3,-2,-1
6	-1 -1 1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 12,2,-12,-2,-8,-10,-4,-6,0,2,8,10,4,-2,-4,-2	1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 1,12,1,0,1,0,-1,2,1,0,-1,0,-1
7	1 -1 1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 8,2,-12,-2,-8,-10,-4,-6,0,2,8,10,4,-2,-4,-2	-1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 -1,12,1,0,1,0,-1,2,1,0,-1,0,-1

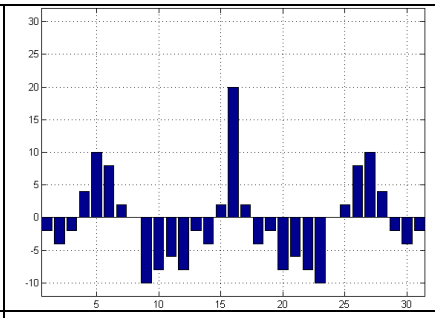
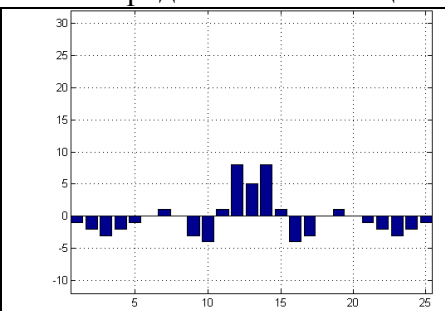
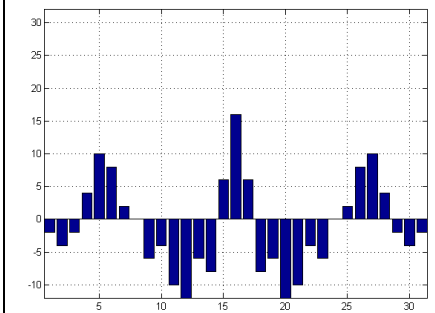
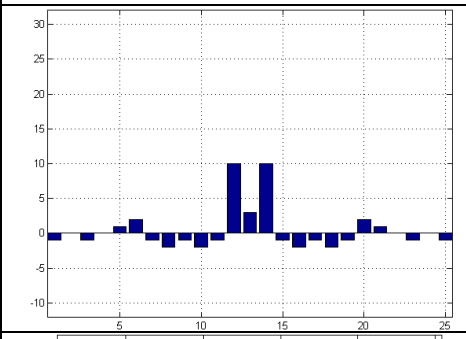
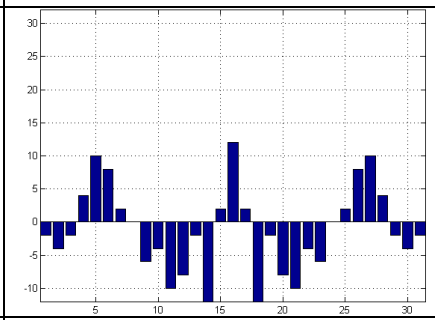
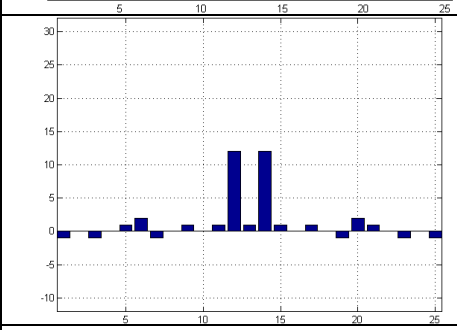
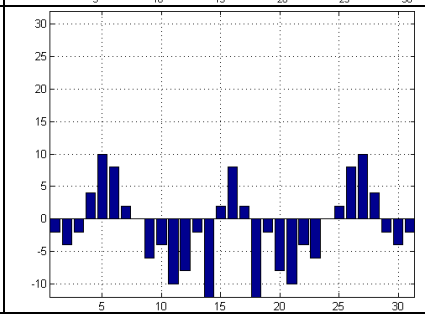
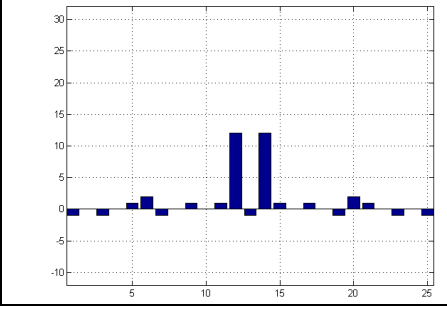
Обмеження числа помилок в досліджуваних кодах визначається до границі втрати кодами особливих кореляційних властивостей, коли величина максимальної бокової пелюстки перевищує величину головної пелюстки, тобто $L_0<L_{i\ max}$, і не виконується умова критерію 13.8.

Таблиця 13.10.

Графіки кореляційних характеристик ШКП при впливі помилок

К-сть помилок	Двовимірний ШКП	Одновимірний код Баркера
1	2	3
0		
1		
2		
3		

Продовження таблиці 13.10

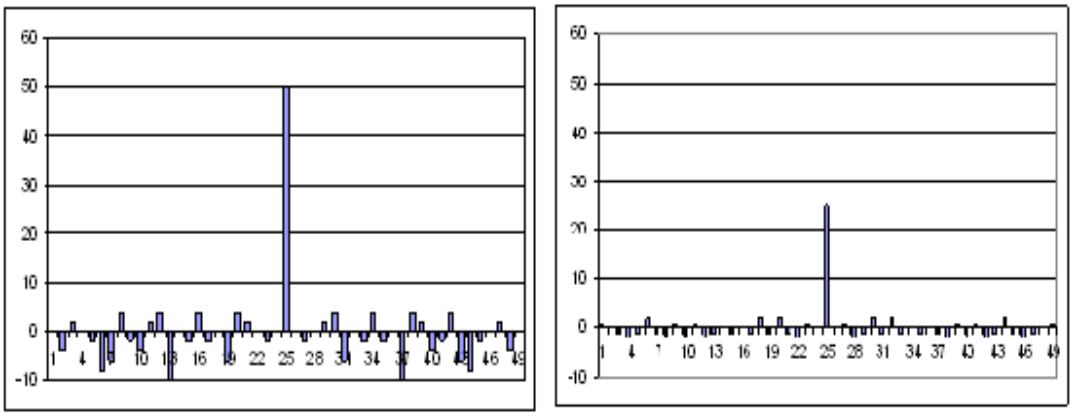
4		
5		
6		
7		

З табл.13.9 та табл.13.10 видно, що при введені помилок в код ШКП і виконанні взаємокореляційної функції з еталоном ШКП двовимірні коди характеризуються суттєво кращими характеристиками по відношенню до одновимірних, що проілюстровано в табл.13.10 при чотирьохкратній помилці.

Для перевірки даного ефекту виконаємо відповідне порівняння велико розрядних кодів однакової довжини двовимірного з розмірністю матриці 5×5 та 25-ти розрядного одновимірного модифікованого коду Баркера:

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{matrix}, \qquad 1100011111110101001001101.$$

На рис.13.20 показані графіки циклічних автокореляційних функцій базових кодів, взятих для порівняння впливу завад на їх характеристики.



а) б)

Рис.13.20. Графіки автокореляційних функцій двовимірного ШКП та одновимірного кодів ШКП однакової розрядності.

З рис.13.20 видно, що при відсутності помилок в ШКП величина максимальних бокових пелюсток одновимірного коду знаходиться в границях ± 2, а в двовимірному ШКП – відповідно +4, -10. При цьому головна пелюстка двовимірного коду в 2 рази перевищує рівень головної пелюстки одновимірного коду. Величина від'ємних значень бокових пелюсток у двовимірному коді не впливає на характеристики приймання та виявлення ШКП, оскільки приймання ведеться згідно принципу амплітудної модуляції.

В табл.13.11 приведені графіки результату аналізу впливу помилок в досліджуваних кодах на їх кореляційні характеристики.

З графіків табл.13.11 видно, що дані коди втрачають особливі кореляційні властивості при шестикратній помилці, що характеризує їх стійкість до впливу завад порядку 24% до довжини коду, для малоймовірних

найбільш вразливих помилок в кодах, які їх спотворюють. Оскільки ймовірність появи помилки z кратності має рівномірний розподіл і описується мультиплікативною функцією, то ймовірність втрати кодом особливих кореляційних властивостей можна розрахувати за формулою:

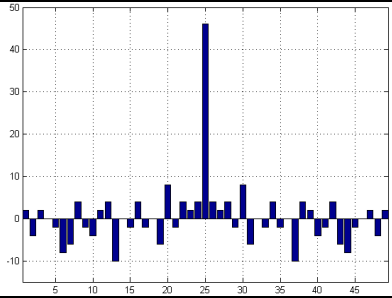
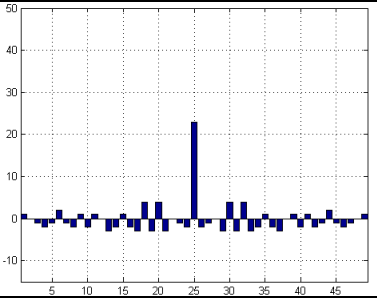
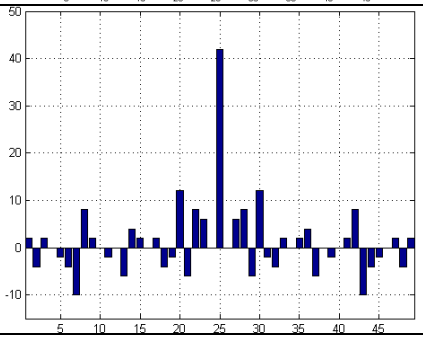
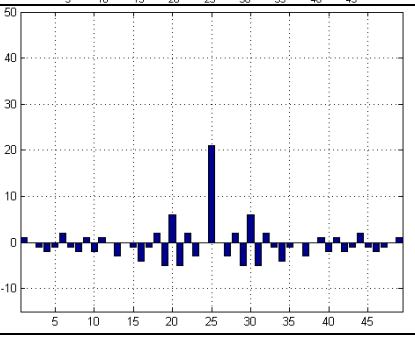
$$P_{ШКП} = \prod_{i=1}^{z} P_i ,$$

тобто для даного коду гранична ймовірність повної втрати його кореляційних характеристик рівна $0{,}24^6$.

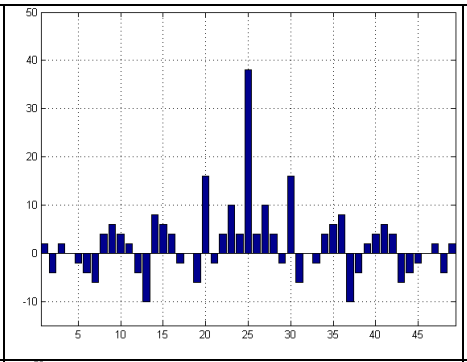
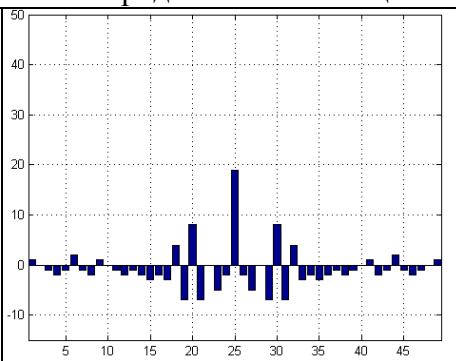
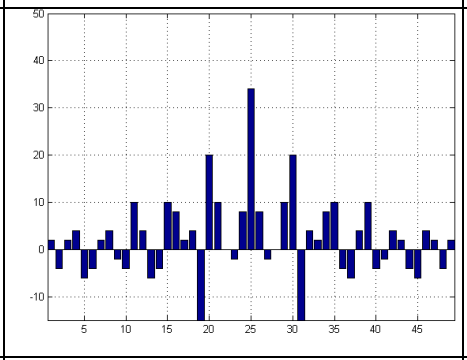
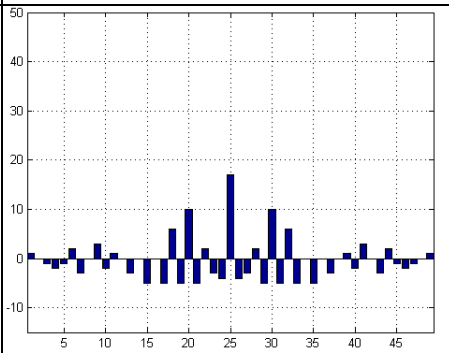
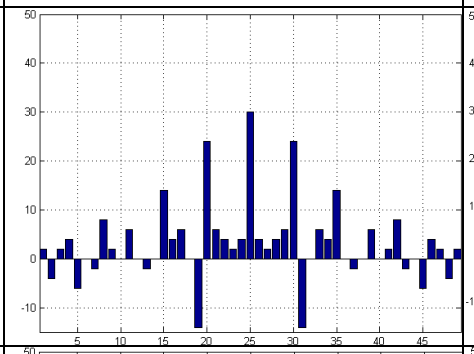
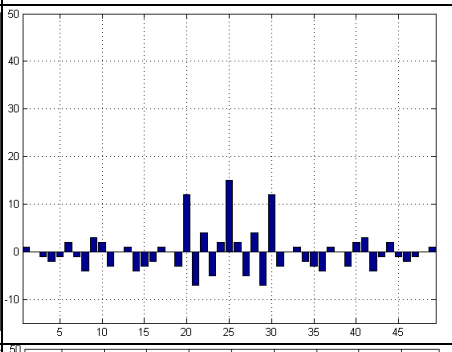
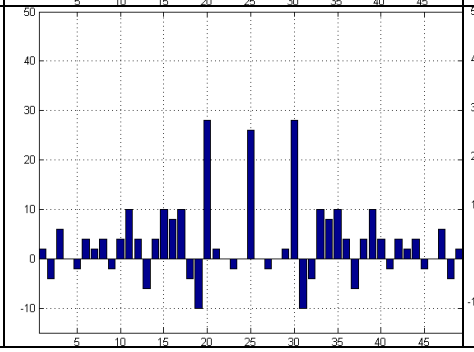
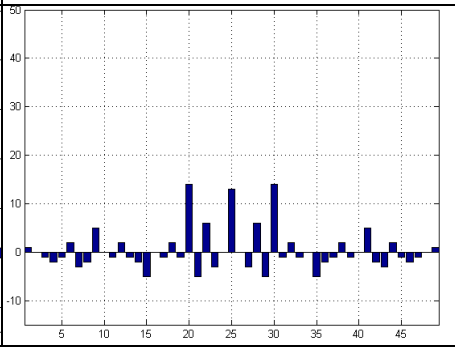
На рис.13.21 показані характеристики зниження кореляційних властивостей досліджуваних кодів в залежності від кратності помилок та критерію ефективності (13.8).

Таблиця 13.11.

Графіки кореляційних характеристик великорозрядних одновимірного та двовимірного ШКП

К-сть п-к	Двовимірний ШКП	Одновимірний ШКП
1		
2		

продовження таблиці 13.11.

3		
4		
5		
6		

Проведений аналіз впливу помилок, які можуть виникати в ШКП внаслідок дії завад в системах обміну даними між віддаленими процесорами,

результати якого приведені на рис.13.21, показує, що в залежності від числа пошкоджених бітів коди з більшою розрядністю характеризуються кращими характеристиками, а розроблені двовимірні ШКП відрізняються стійкою перевагою над іншими типами відомих кодів даного класу.

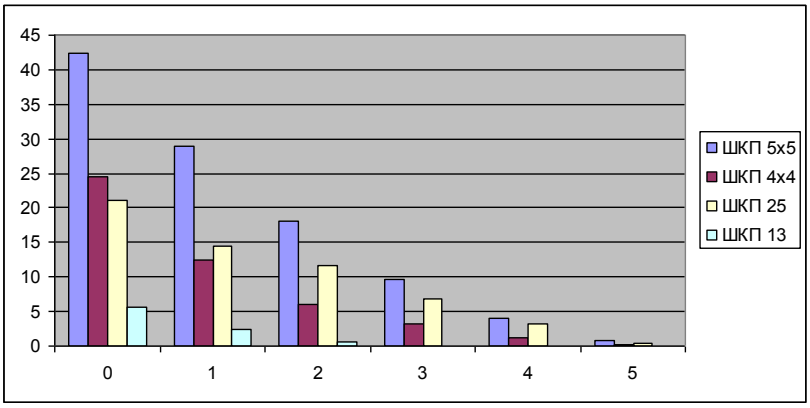


Рис.13.21. Характеристики ефективності ШКП в залежності від кратності помилок.

Проведені результати моделювання дослідження кореляційних характеристик різних типів одновимірних та запропонованих двовимірних ШКП, як це видно з рис.13.19, демонструють тенденції покращення системних характеристик даних кодів при зростанні їх розрядності. При цьому необхідно задіювати великі мультипроцесорні ресурси для знаходження ефективних кодових послідовностей при збільшенні їх розрядності в границях 32-64 біт. В даному випадку запропонований критерій (13.4) дозволяє знайти найоптимальніші двовимірні коди з найкращими кореляційними властивостями. Слід зауважити, що в процесі аналізу виявлені окремі факти симетрії в кращих двовимірних кодах, а також наявності вставлених М-послідовностей, що є еталонними двовимірними кодами (рис.13.22).

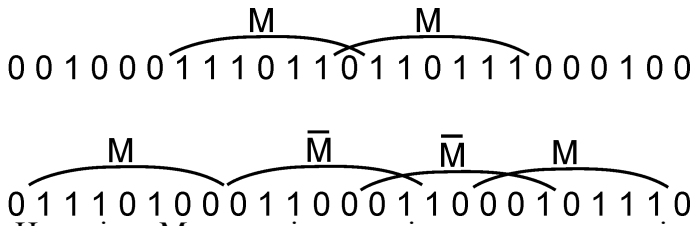


Рис.13.22. Наявність М-сигналів в досліджуваних двовимірних кодах.

Виявлені властивості симетрії та наявності фрагментів одновимірних М-послідовностей в кращих двовимірних ШКП приводить до висновку, що

можуть існувати більш фундаментальні теоретико-числові основи побудови даного класу кодів і прискорення швидкості їх пошуку на числовій осі.

На рис.13.23 зображено графічну інтерпретацію двовимірних ШКП розрядностей 3х3 (рис.13.23а), 4х4(рис.13.23б), 5х5 (рис.13.23в).

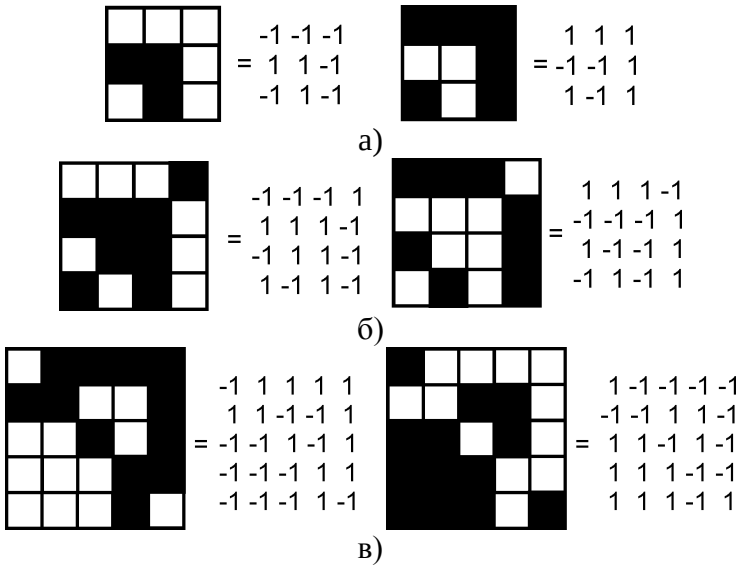


Рис.13.23. Графічні інтерпритації кодових матриць двомірних ШКП.

При графічній інтерпритації кодових матриць знайдених двовимірних ШКП видно симетричність їх матриць.

Отже, двовимірні ШКП з хорошими кореляційними властивостями мають яскраво виражену симетричність, що дозволяє їх оцінювати візуально. Можна висунути припущення, що при подальшому дослідженні даних кодових послідовностей та розвитку їх теорії в напрямку побудови тривимірних ДІ з особливими кореляційними властивостями. Адже вони також можуть мати симетричні властивості, а це дозволяє значно спростити алгоритми пошуку ефективних ШКП великої розрядності.

РОЗДІЛ 14

СИГНАЛЬНІ КОРЕКТУЮЧІ КОДИ В БАЗИСІ ГАЛУА

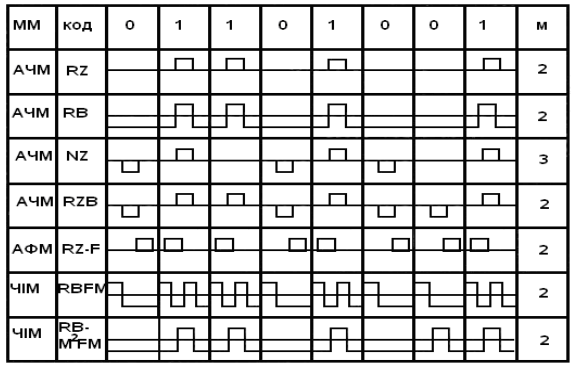
14.1. Методи маніпуляції сигналів на основі дискретних кусково-постійних функцій.

В сучасних комп'ютерних системах з оптичним ІК найширше застосування отримали двомірні методи імпульсної, а в окремих випадках - потенціальної маніпуляції.

Імпульсні методи маніпуляції сигналів найчастіше використовують на низових рівнях комп'ютерних мереж, в цифровій телефонії, а також існуючих комп'ютерних системах з оптичними ІК. У зв'язку з тим, що дані методи використовують обмежену частину енергії на інтервалі тривалості сигналу, а також потребують широкої смуги частот в каналі зв'язку, ефективність їх недостатньо висока. Серед цих методів застосовують імпульсну та диференційно-імпульсно кодову маніпуляцію (табл.14.1).

Таблиця 14.1.

Методи одновимірної імпульсної маніпуляції.



Порівняльний аналіз методів ефективності (переваг та недоліків) різних методів двовимірної імпульсної маніпуляції сигналів поданий у (табл.14.2).

Таблиця 14.2.

Класифікація переваг та недоліків різних методів двомірної маніпуляції.

Метод маніпуляції	Переваги	Недоліки
АЧМ (RZ)	простий закон маніпуляції, сигнали однополярні	низька енергія сигналу, відсутня самосинхронізація.

Продовженя таблиці 14.2.

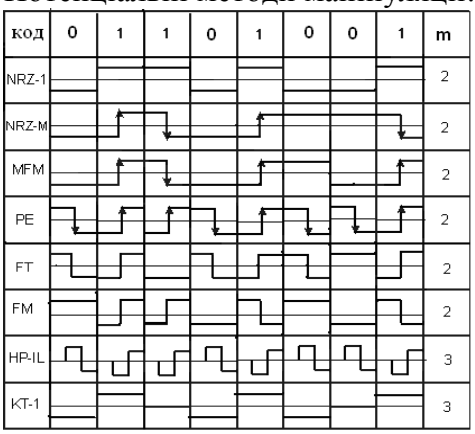
АЧМ (RB)	Вдвічі більша енергія сигналу, по відношенню до методу RZ	сигнал знакозмінний; загальний недолік для RZ і RB дрейф нуля в лінії зв'язку
АЧМ (NZ)	Більш високий рівень серії бітової синхронізації	сигнал знакозмінний
АЧМ (RZB)	синхронізація кожного біту, що передається	Загальним недоліком методів RZB, RZ-F, RB-FM, RB-M^2FM є використання малої кількості енергії сигнального простору.
АФМ (RZ-F)	самосинхронізація кожного біту даних	
ЧІМ (RBFM)	самосинхронізація та завадозахищеність	
ЧІМ (RB-M^2 FM)		

Загальним недоліком всіх представлених одновимірних методів маніпуляції є широкий спектр сигналів, а також неефективне використання сигнального вікна, що приводить до зниження швидкості передавання інформації.

Широкого застосування в сучасних комп'ютерних системах набули також потенціальні методи маніпуляції сигналу. Дані методи характеризуються підвищеною енергією сигналів, що передаються. Вони бувають двох типів: без самосинхронізації (потребують додаткового каналу синхронізації) і з самосинхронізацією (табл.14.3, 14.4).

Таблиця 14.3.

Потенціальні методи маніпуляції.



Серед перелічених методів, метод КТ-1 є найбільш ефективним, оскільки володіє наступними позитивними якостями:

1. Виключає повторення однакових ознак сигналів і забезпечує якісну бітову синхронізацію.
2. Двократні повторення сигналів використовуються в якості start-stop сигналів, що виключає необхідність формування флагів на границях інформаційних масивів і біт-стаффінгів в середині інформаційного масиву.

Метод КТ -1 розроблений і успішно застосований в низових рівнях комп'ютерних систем у нафтогазовій галузі.

Таблиця 14.4.

Порівняльні характеристики різних методів потенційних методів. маніпуляції

Метод маніпуляції	Переваги	Недоліки
NRZ – 1	забезпечує максимальну енергію сигналів, але відсутня бітова синхронізація	дрейф нуля в лінії зв'язку
NRZ – M	забезпечує синхронізацію сигналів "1"	
MFM	забезпечує фронтом наростанням або наростання нулів і передачу потенціалу	синхронізація бітів фронтами наростання та спаду сигналу з фазовою маніпуляцією
PE	нуль передається фронтом спаду, а одиничка фронтом наростання	неефективна бітова синхронізація
FT	висока завадозахищеність і самосинхронізація	зміна швидкості передавання даних, дрейф нуля
HP – IL	характеризується самосинхронізацією, відсутністю дрейфу нуля	низька швидкість передачі даних при заданій смузі частот каналу зв'язку
КТ – 1	максимальна швидкість передачі сигналів і самосинхронізація	дрейф нуля. Для ліквідації дрейфу сигнали "0", "1" і сигнал "повторення" передаються трьома частотами.

У (табл.14.5). показані приклади маніпуляції сигналів модифікованими методами квазітрійкової маніпуляції, які характеризуються відсутністю дрейфу нуля.

Таблиця 14.5.

Формування імпульсної послідовності методом КТ-М.

код	0	1	1	0	1	0	0	1	m
FM-KT1									З
КТ-М									З

Загальним недоліком проаналізованих методів маніпуляції є широкий спектр сигналів, як наслідок цього, відбувається зниження швидкості передавання інформації і високий рівень завад між каналом зв'язку. Тому доцільно використовувати методи квазітрійкової маніпуляції сигналів покращеної форми з вузьким спектром (табл.14.6).

Таблиця 14.6.

Квазітрійкові методи маніпуляції сигналів покращеної форми.

ФОРМА сигналу	Метод	Бітова послідовність							
		0	1	1	0	1	0	0	1
Трапеціедальні	КТ-Т	+1 -1							
Дзвоноподібні $\sin^2 x$	КТ-D	+1 -1							
$\frac{\sin x}{x}$	КТ-S	+1 -1							

Ефективність методу маніпуляції сигналу також залежить від реалізації блокової синхронізації.

При цьому аналіз ефективності методу маніпуляції сигналів з врахуванням блокової синхронізації можна обчислити за допомогою наступного виразу:

$$K_e = (n+m)/n \qquad (14.1)$$

де n – число біт-інформаційної кодової послідовності, m – число біт коду блокової синхронізації.

На (рис.14.1) наведено приклад виконання блокової синхронізації при використанні стандартного протоколу обміну даними в комп'ютерних системах HDLC, методу маніпуляції HP-IL та КТ-1 і без врахування коефіцієнта форми сигналу та ступеня використання енергії сигнального вікна.

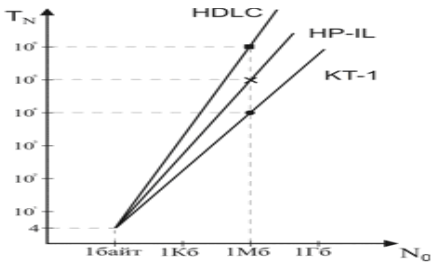


Рис.14.1. Характеристика швидкості передавання даних в стандартних протоколах:

N_0 – об'єм даних; T_N – швидкість передавання в залежності від об'єму даних.

Серед методів маніпуляції найбільш широко використовується Манчестерський код (рис.14.2).

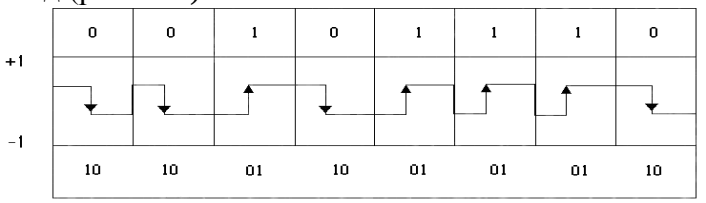


Рис.14.2. Кодування сигналу на основі Манчестерського коду.

Аналіз (рис.14.2.) показує, що незважаючи на те, що при передаванні байту даних (8 біт), в каналі є присутні ще 2 байти двійкових сигналів - цей факт пояснюється тим, що для передавання Манчестерським кодом " 1" і "0" використовуються не всі позиції сигнального простору. З (рис.14.3.) видно, що дві позиції "00" і "11" в каналах зв'язку фактично не використовуються.

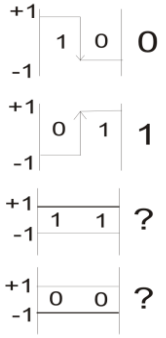


Рис.14.3. Представлення кодової послідовності з використанням бітів даних.

Існують інші методи маніпуляції, де ці коди використовують для повторення нулів і одиниць, що підвищує ефективність передачі інформації [83] (рис.14.4).

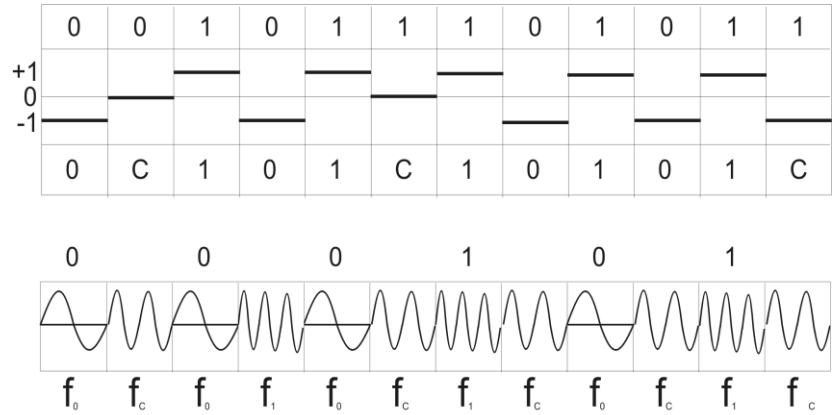


Рис.14.4 Метод кодування КТ-F.

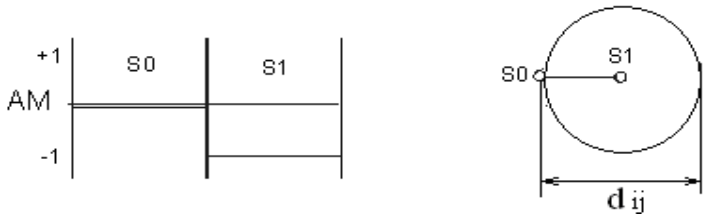
C – синхронізація; f_0, f_c , f_1 – відповідні частотина яких передаються нулі та одиниці.

Всі відомі методи маніпуляції в принципі не дозволяють виявити помилки в каналах зв'язку без введення надлишкової інформації. Застосування кодів Галуа дозволяє створити нові методи маніпуляції сигналів з можливим виправленням і виявленням помилок без введення кодової надлишковості.

14.2. Розрахунок сигнальних просторів багаторівневої маніпуляції функціями різних ТЧБ.

Для порівняння ефективності різних методів маніпуляції необхідно виконати їх розрахунок на основі сигнальних моделей.

Представлення методів маніпуляції відбувається в одиничному колі (рис.14.5).



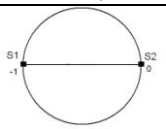
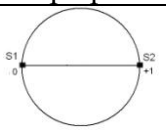
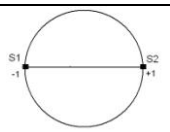
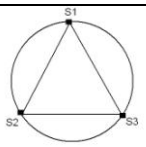
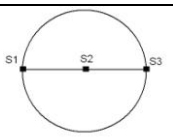
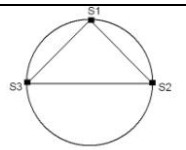
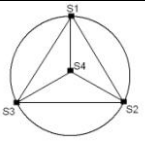
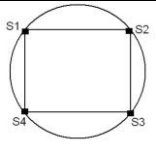
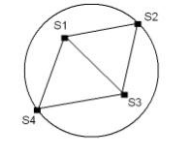
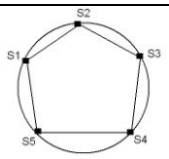
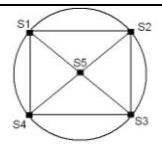
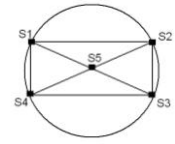
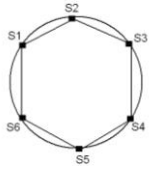
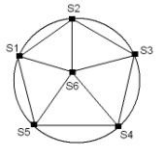
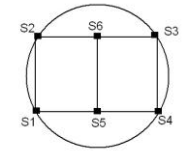
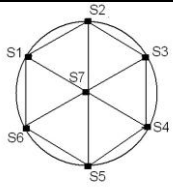
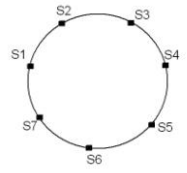
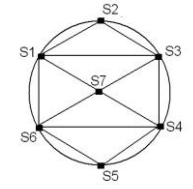
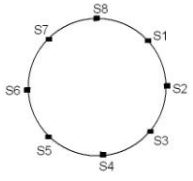
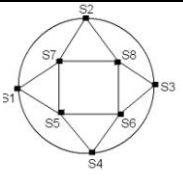
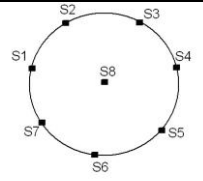
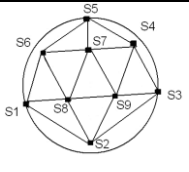
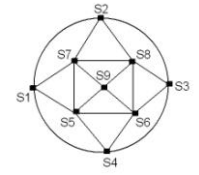
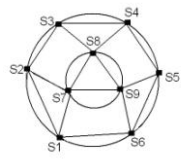
а) б)

Рис.14.5 Сигнальне коло при амплітудній (а) та фазовій маніпуляціях (б):S_i S_j - сигнали в сигнальному просторі; d_{ij} - кодова віддаль; Ei = R_i , E - енергія сигналу.

Чим більше d_{ij}, тим ефективність методу маніпуляції вища (табл.14.7).

Таблиця 14.7.

Систематизація сигнальних кіл при різних ознаках сигнального вікна

M=2			
M=3			
M=4			
M=5			
M=6			
M=7			
M=8			
M=9			

M* – число ознак сигнального вікна (число маніпуляцій).

Ефективність різних методів маніпуляції можна оцінити на основі виразу:

$$K_{ем} = \frac{k_{\Phi} \cdot dij\min \cdot \sqrt{Log_2 M}}{2\sqrt{Ec}} \qquad (14.2)$$

де d_{ij} - кодова віддаль між маніпульованими сигналами;
M - число ознак маніпуляції;
E_c - середня енергія сигналу;
$K_ф$ - коефіцієнт форми сигналу (залежить від базисної функції);
S_i - енергія сигналів в двомірному сигнальному просторі.

E_c розраховується по формулі:

$$E_c = \frac{1}{M} \sum_{i=1}^{M} S_i \qquad (14.3)$$

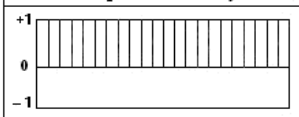
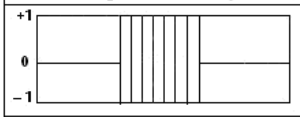
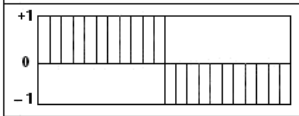
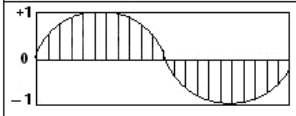
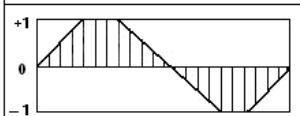
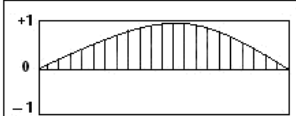
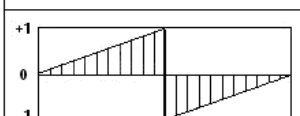
В оптичних каналах використовуються виключно двохрівневі сигнали S_0= 0; S_1 = +1. Тому d_{ij} = 1; E_c= 0+1/2 = 0.5.

Незважаючи на різноманітність даних сигналів, всі вони кодуються в сигнальному вікні з кодовою віддаллю $d_{ij} = 1$, з коефіцієнтом форми $K_f = 0{,}5$. Використання даного типу сигналів на інтервалі одного сигнального вікна дозволяє збільшити число M (число маніпуляцій) і тим самим підвищити швидкість передавання даних по оптичних каналах зв'язку. Ефективність методів маніпуляції $K_{еф}$ залежить лише від форми сигналів K_f.

Дані аналізу ефективності методів маніпуляції в залежності від форми сигнального вікна представлено у (табл.14.8).

Таблиця 14.8.

Сигнальні вікна оптичних сигналів різних типів маніпуляції.

Форма сигналу	$K_{еф}$	Форма сигналу	$K_{еф}$
	1		0.6
	1		0.6
	0.7		0.9
	0.7		0.5

Для підвищення завадозахищеності оптичних каналів зв'язку необхідно використовувати широкосмугові сигнали на основі наступних випадкових послідовностей теоретикио-числового базисів Галуа:

- ШПС (шумоподібні сигнали);
- М-послідовності (послідовності довжини максимуму);
- коди Баркера;
- модифіковані коди Баркера ;
- двомірні коди Баркера.

14.3. Оцінка надлишковості захисту даних від помилок існуючих протоколів передавання даних.

Для виявлення помилок при передаванні сигналу використовують стандартні методи на основі рекурентних надлишкових коректуючих кодів .

Для того, щоб в даних не з'являвся код флага після кожних п'яти одиниць, виконується процедура біт-стаффінга, яка потім вилучається. В результаті код-фрейм має змінну довжину, що ускладнює процедуру виявлення помилок і перевантажує трафік передачі даних. На (рис.14.6) показана структура фрейму стандартного протоколу HDLC, рекомендації МККТ х.25.

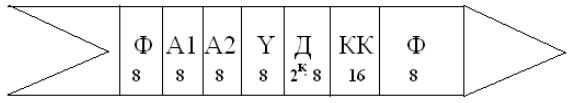


Рис.14.6. Структура фрейму протокола HDLC:
Ф – флаг (байт 01111110); А1,А2 – відповідно адреси передавальної та приймальної станцій; Y – тип кадру; Д – дані ($Д=2^K \cdot 8$);

КК – коректуючий код БЧХ.

Для дослідження надлишковості, що виникає при передаванні даних, згідно протоколу розглянемо фрагмент вхідних даних (рис.14.7), який внаслідок виникнення операції біт-стаффінга передається по інформаційному каналу у вигляді коду: 01111110.001111101.111110…(рис.2.13).

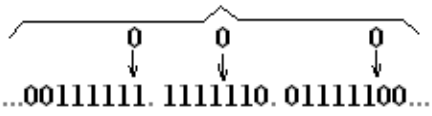


Рис.14.7. Фрагмент вхідних даних.

Таким чином, при використанні названого протоколу виникає надлишковість обумовлена необхідністю додаткового передавання службової інформації у вигляді флагів захисного коректуючого коду та певного числа вставок нулів біт-стаффінга.

В загальному випадку надлишковість даних, які передаються згідно описаного протоколу, можна оцінити у вигляді коефіцієнта надлишковості як відношення об'єму даних, що передаються, до об'єму вхідних даних, які підлягають передаванню, зідно аналітичних виразів :

$$К_н = V_k / V_x; \qquad (14.4)$$
$$V_k = V_c + V_x; \quad V_c = V_Ф + V_A + V_Y + V_{kk} + V_Б ,$$

де $V_Ф$, V_A, V_Y , V_{kk}, $V_Б$ – відповідні об'єми кодів флага, адреса станцій, типу фрейму та коректуючого коду БЧХ, символів біт-стаффінга; V_k – об'єм даних, що передаються в ІК, V_c – об'єм службових даних, V_x – об'єм вхідних даних.

Оскільки масиви даних передаються у вигляді блоків $2^k \cdot 8$ біт, де k=1,2,…,n, а граничний об'єм надлишковості даних біт-стаффінга оцінюється величиною 2^b , де b = 0,1,2,…,m , отримаємо загальну формулу надлишковості існуючого стандартного методу передавання даних :

$$К_{н\ (HDLC)} = (V_Ф + V_A + V_Y + V_{kk} + V_Б + V_x) / К_{ФС} \cdot V_x ,$$
$$К_{н\ (HDLC)} = (V_Ф + V_A + V_Y + V_{kk} + 2^b + 2^k \cdot 8) / 2^k \cdot 8. \qquad (14.5)$$

$К_{ФС}$ – коефіцієнт форми сигналу; $К_{ФС\ (HDLC)} = 0.5$.

На рис.2.14 показано фрейм протоколу HP-IL (Hawlett Packard), орієнтованого на низові рівні РКС.

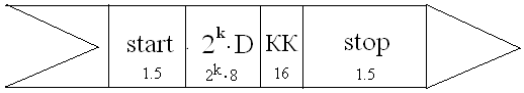


Рис.14.8. Структура фрейму протоколу HP-IL.

Виходячи із структури даного протоколу, оцінка надлишковості передавання даних розраховується згідно виразу :

$$К_{н\ (HP-IL)} = (3 + 2^k \cdot 8 + V_{kk}) / К_{ФС} \cdot 2^k \cdot 8 , \qquad (14.6)$$

де $К_{ФС(HP-IL)} = 0.3$; $V_{kk} = 16$.

Структуру фрейму протоколу КТ-1 представлено на (рис.14.9).

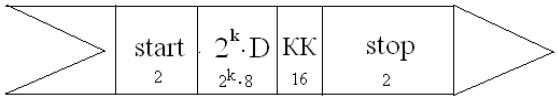


Рис.14.9. Структура фрейму протоколу КТ-1.

Згідно (рис.14.9) оцінка надлишковості передавання даних розраховується на основі виразу :

$$К_{н\ (КТ\text{-}1)} = (4 + 2^k \cdot 8 + V_{kk}) / К_{ФС} \cdot 2^k \cdot 8 , \qquad (14.7)$$

де $К_{ФС\ (КТ\text{-}1)} = 1.0$; $V_{kk} = 16$.

На (рис.14.10) показано графік надлишковості передавання даних існуючими методами та протоколами.

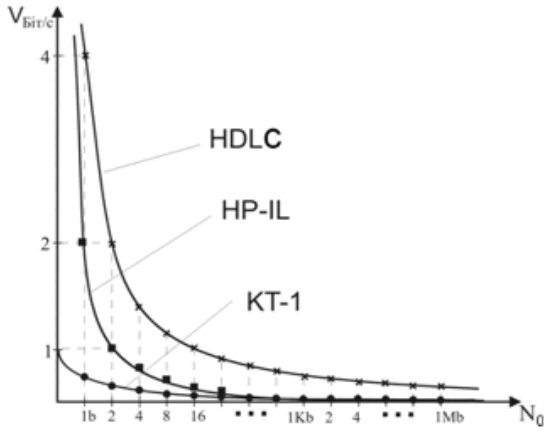


Рис.14.10. Графік надлишковості передавання даних протоколами HDLC, HP-IL, KT-1.

Аналіз (рис.14.10) показує, що надлишковість існуючих протоколів різко зростає при малих об’ємах даних, які передаються, що характерно для низових рівнів РКС. Тому їх застосування на різних рівнях сучасних комп’ютерних систем є недостатньо ефективне і потребує вдосконалення як в теоретичному, так і в практичному планах.

14.4. Методи безнадлишкового сигнального кодування на основі кодів Галуа.

Поняття безнадлишкового сигнального кодування базується на принципі створення кодів з можливістю виявлення та виправлення помилок, які не призводять до збільшення числа сигналів при передаванні біт-орієнтованих потоків даних.

Суть методів безнадлишкового сигнального кодування з можливістювиявлення та виправлення помилок полягає в тому, що при формуванні такого класу кодів використовується до п’яти сигнальних ознак наступного типу:

- фронт наростання _↑‾ (∧) ;

- фронт спаду ‾↓_ (∨) ;
- додатній потенціал |‾‾| (+);
- від'ємний потенціал |__| (–);
- нульовий потенціал |▬| (S);

Зауважимо, що всі ці ознаки різними способами використовуються в стандартних потенціальних методах маніпуляції (див. табл.14.6), при чому послідовність бітів даних додатково модулюється кодом поля Галуа.

Розроблено чотири можливих способи формування такого класу кодів:

- позиційно-сигнальний код (ПСК);
- несиметричний рекурентний сигнальний код (НРСК);
- рекурентний симетричний сигнальний код (РССК);
- квазі-символьний сигнальний код (КССК).

14.4.1. Метод формування позиційно-сигнального коду (ПСК).

Введемо наступну систему представлення сигнальних та кодових ПСК:

$$G_0^{\,1} \Rightarrow «+» \; ; \; G_0^{\,0} \Rightarrow «–» \; ; \; G_1^{\,1} \Rightarrow «\wedge» \; ; \; G_1^{\,0} \Rightarrow «\vee».$$

Біт «нуль», який на сигнальному рівні кодується в Галуа одиницею, представляється сигналом +1 і відповідним символом «+» в кодовому вигляді. Біт «нуль», який кодується в Галуа нулем, на сигнальному рівні, представлено потенціалом -1 , а в кодовому вигляді відповідним символом «–» .

Біт «одиниця», яка кодується в Галуа одиницею, представляється фронтом наростання на сигнальному рівні або символом ∧ . Біт «одинииця», яка кодується в Галуа символом нуль, представляється фронтом спаду на сигнальному рівні або символом ∨ в кодовому вигляді.

Формування ПСК показано в (табл.14.9).

На (рис.14.11) представлена символіка розрядно-позиційного сигнального коду Галуа для півбайтових потоків даних, що відповідає потоку цифр.

1	**1**	**1**	**1**		**0**	**0**	**0**	**0**
G1	G1	G0	G0		G1	G1	G0	G0
↑	↑	↓	↓		+1	+1	-1	-1
∧	∧	∨	∨		+	+	–	–

Рис.14.11. Представлення кодів ПСК на сигнальному рівні.

При формуванні ПСК даним методом для виключення появи в кодах даних комбінацій Start/Stop – сигналів, останній формується у вигляді трьохбітових послідовностей відповідно: Start – – – Stop +++.

Тобто число start/stop –сигналів повинно розраховуватись згідно виразу:

$$S = n+2 \text{ або } n+6, \qquad (14.8)$$

де n – розрядність кодону Галуа масиву даних, що передаються.

Розглянемо ряд етапів досліджень запропонованого методу ПСК при зміні позиційно-сигнального кодування даних, що передаються в ІК.

Нехай формат даних, що передаються по ІК, рівний 4-м бітам (n=4). Тоді код поля Галуа G_2^2 (формується у вигляді послідовності 1100:

G **1 1 0 0**

start 0 0 0 0 stop

– – – + + – – +++ .

Для 4-х бітового повідомлення, яке передається, існує $m=2^4$ комбінацій ПСК, які в сигнальному та кодовому виді наведені в табл.14.9.

Дослідимо вплив однократних помилок у всіх комбінаціях 4-х бітового ПСК. Фактично, при переході від двійкового коду до ПСК на сигнальному рівні формується четвірковий код, який має об'єм $N= 4^4 = 256$ комбінацій, з яких при передаванні даних використовують $2^4 = 16$ комбінацій.

Таблиця 14.9.

Коректуючі властивості ПСК.

$N_п$	код даних	start			G_1	G_1	G_0	G_0	stop			Кодове представлення даних
0	0000	‾	‾	‾	‾	‾	_	_	_	_	_	– – – + + – – +++
1	0001	‾	‾	‾	‾	‾	_	↓	_	_	_	– – – + + – ∨ +++
2	0010	‾	‾	‾	‾	‾	↓	_	_	_	_	– – – + + ∨ – +++
3	0011	‾	‾	‾	‾	‾	↓	↓	_	_	_	– – – + + ∨ ∨ +++
4	0100	‾	‾	‾	‾	↑	_	_	_	_	_	– – – + ∧ – – +++
5	0101	‾	‾	‾	‾	↑	_	↓	_	_	_	– – – + ∧ – ∨ +++
6	0110	‾	‾	‾	‾	↑	↓	_	_	_	_	– – – + ∧ ∨ – +++
7	0111	‾	‾	‾	‾	↑	↓	↓	_	_	_	– – – + ∧ ∨ ∨ +++
8	1000	‾	‾	‾	↑	‾	_	_	_	_	_	– – – ∧ + – – +++

продовження таблиці14.9.

9	1001	—	—	—	↑	—	_	↓	_	_	_	− − − ∧ + − ∨ + + +
10	1010	—	—	—	↑	—	↓	_	_	_	_	− − − ∧ + ∨ − + + +
11	1011	—	—	—	↑	—	↓	↓	_	_	_	− − − ∧ + ∨ ∨ + + +
12	1100	—	—	—	↑	↑	_	_	_	_	_	− − − ∧ ∧ − − + + +
13	1101	—	—	—	↑	↑	_	↓	_	_	_	− − − ∧ ∧ − ∨ + + +
14	1110	—	—	—	↑	↑	↓	_	_	_	_	− − − ∧ ∧ ∨ − + + +
15	1111	—	—	—	↑	↑	↓	↓	_	_	_	− − − ∧ ∧ ∨ ∨ + + +

Тобто сигнальна надлишковість запропонованого методу на основі ПСК у випадку , коли однократні помилки не призводять до формування дозволених комбінацій, буде дорівнювати 1/16.

Нехай, передається двійковий код «0000», якому відповідає ПСК «++ – –», причому помилка виникає в першому розряді. Тоді під впливом завад в ІК можливе формування трьох кодів із синдромом невиявленої помилки:

1. ∧ + – – ■ + – –,
2. ∨ + – – ⇒
3. – + – –

де символ ■ – відповідає можливій появі символу ∧. При цьому однозначно виявляється помилка в першому розряді, типу ∨ +, оскільки після старту не існує правильної комбінації, яка не містить двох послідовних символів + + або двох символів ∧ ∧ . Аналогічно виявляється помилка при наявності комбінації « – + ».

Введемо помилку в другому розряді 4-х бітового коду даних:

1. + ∧ – – + ■ – –
2. + ∨ – – ⇒
3. + – – –

Оскільки двох таких стартових комбінацій нулів в базисі Галуа не існує, то помилка однозначно не виявляється лише в одному випадку.

Введемо помилку в третьому розряді даних:

1. + + ∧ –
2. + + ∨ – ⇒ + + ■ –
3. + + + –

В третьому розряді даним ПСК помилка не виявляється теж в одному випадку. Введемо помилку в четвертому розряді:

1. + + – ∧
2. + + – ∨ ⇒ + + – ■
3. + + – +

Аналогічно, як і в попередньому випадку, не виявляється одна помилка.

Результати досліджень можливості виявлення однократних помилок для всіх кодових комбінацій даних від 0000 до 1111 приведені в (табл. 14.10).

Таблиця 14.10.

Коректуючі властивості ПСК для 4-х біт даних.

$N_п$	Код даних	Код ПСК	Помилка в **1**-му розряді	Помилка в **2**-му розряді	Помилка в **3**-му розряді	Помилка в **4**-му розряді
0	0000	+ + – –	■ + – – ∨ + – – – + – –	+ ■ – – + ∨ – – + – – –	+ + ∧ – + + ■ – + + + –	+ + – ∧ + + – ■ + + – +
1	0001	+ + – ∨	■ + – ∨ ∨ + – ∨ – + – ∨	+ ■ – ∨ + ∨ – ∨ + – – ∨	+ + ∧ ∨ + + ■ ∨ + + + ∨	+ + – ∧ + + – ■ + + – +
2	0010	+ + ∨ –	■ + ∨ – ∨ + ∨ – – + ∨ –	+ ■ ∨ – + ∨ ∨ – + – ∨ –	+ + ∧ – + + ■ – + + + –	+ + ∨ ∧ + + ∨ ■ + + ∨ +
3	0011	+ + ∨ ∨	■ + ∨ ∨ ∨ + ∨ ∨ – + ∨ ∨	+ ■ ∨ ∨ + ∨ ∨ ∨ + – ∨ ∨	+ + ∧ ∨ + + ■ ∨ + + + ∨	+ + ∨ ∧ + + ∨ ■ + + ∨ +
4	0100	+ ∧ – –	■ ∧ – – ∨ ∧ – – – ∧ – –	+ ∨ – – + ■ – – + – – –	+ ∧ ∧ – + ∧ ■ – + ∧ + –	+ ∧ – ∧ + ∧ – ■ + ∧ – +
5	0101	+ ∧ – ∨	■ ∧ – ∨ ∨ ∧ – ∨ – ∧ – ∨	+ ∨ – ∨ + ■ – ∨ + – – ∨	+ ∧ ∧ ∨ + ∧ ■ ∨ + ∧ + ∨	+ ∧ – ∧ + ∧ – + + ∧ – ■
6	0110	+ ∧ ∨ –	■ ∧ ∨ – ∨ ∧ ∨ – – ∧ ∨ –	+ ∨ ∨ – + ■ ∨ – + – ∨ –	+ ∧ ∧ – + ∧ + – + ∧ ■ –	+ ∧ ∨ ∧ + ∧ ∨ ■ + ∧ ∨ +
7	0111	+ ∧ ∨ ∨	■ ∧ ∨ ∨ ∨ ∧ ∨ ∨ – ∧ ∨ ∨	+ ∨ ∨ ∨ + ■ ∨ ∨ + – ∨ ∨	+ ∧ ∧ ∨ + ∧ + ∨ + ∧ ■ ∨	+ ∧ ∨ ∧ + ∧ ∨ + + ∧ ∨ ■
8	1000	∧ + – –	∨ + – – ■ + – – – + – –	∧ ■ – – ∧ ∨ – – ∧ – – –	∧ + ∧ – ∧ + ■ – ∧ + + –	∧ + – ∧ ∧ + – ■ ∧ + – +

Продовження таблиці 14.10

9	1001	∧ + − ∨	∨ + − ∨ ■ + − ∨ − + − ∨	∧ ■ − ∨ ∧ ∨ − ∨ ∧ − − ∨	∧ + ∧ ∨ ∧ + ■ ∨ ∧ + + ∨	∧ + − ∧ ∧ + − + ∧ + − ■
10	1010	∧ + ∨ −	∨ + ∨ − ■ + ∨ − − + ∨ −	∧ ■ ∨ − ∧ ∨ ∨ − ∧ − ∨ −	∧ + ∧ − ∧ + + − ∧ + ■ −	∧ + ∨ ∧ ∧ + ∨ ■ ∧ + ∨ +
11	1011	∧ + ∨ ∨	∨ + ∨ ∨ ■ + ∨ ∨ − + ∨ ∨	∧ ■ ∨ ∨ ∧ ∨ ∨ ∨ ∧ − ∨ ∨	∧ + ∧ ∨ ∧ + + ∨ ∧ + ■ ∨	∧ + ∨ ∧ ∧ + ∨ ■ ∧ + ∨ +
12	1100	∧ ∧ − −	∨ ∧ − − ■ ∧ − − − ∧ − −	∧ ∨ − − ∧ ■ − − ∧ − − −	∧ ∧ ∧ − ∧ ∧ ■ − ∧ ∧ + −	∧ ∧ − ∧ ∧ ∧ − ■ ∧ ∧ − +
13	1101	∧ ∧ − ∨	∨ ∧ − ∨ ■ ∧ − ∨ − ∧ − ∨	∧ ∨ − ∨ ∧ ■ − ∨ ∧ − − ∨	∧ ∧ ∧ ∨ ∧ ∧ ■ ∨ ∧ ∧ + ∨	∧ ∧ − ∧ ∧ ∧ − + ∧ ∧ − ■
14	1110	∧ ∧ ∨ −	∨ ∧ ∨ − ■ ∧ ∨ − − ∧ ∨ −	∧ ∨ ∨ − ∧ ■ ∨ − ∧ − ∨ −	∧ ∧ ∧ − ∧ ∧ + − ∧ ∧ ■ −	∧ ∧ ∨ ∧ ∧ ∧ ∨ ■ ∧ ∧ ∨ +
15	1111	∧ ∧ ∨ ∨	∨ ∧ ∨ ∨ ■ ∧ ∨ ∨ − ∧ ∨ ∨	∧ ∨ ∨ ∨ ∧ ■ ∨ ∨ ∧ − ∨ ∨	∧ ∧ ∧ ∨ ∧ ∧ + ∨ ∧ ∧ ■ ∨	∧ ∧ ∨ ∧ ∧ ∧ ∨ + ∧ ∧ ∨ ■

В результаті аналізу (табл.14.15) побудовані графи можливих кодових переходів при виявленні помилок в різних розрядах кодів, які представлено на рис. (14.11-14.14).

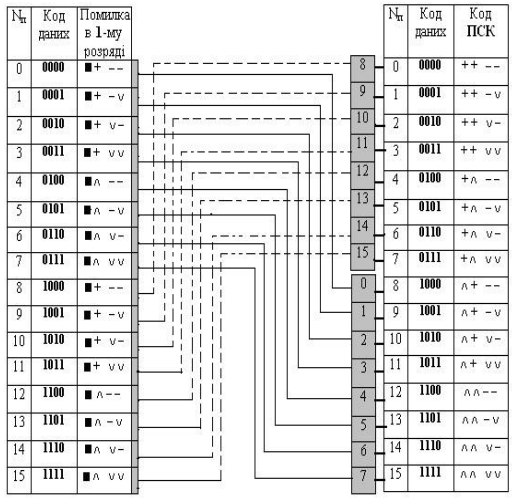


Рис.14.11. Представлення невиявленої помилки в першому розряді кодів ПСК.

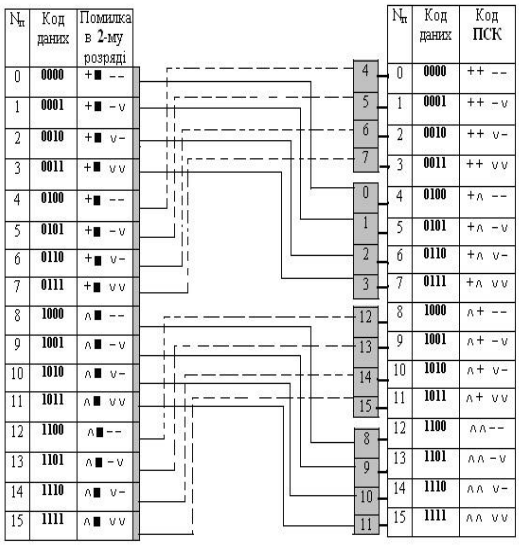


Рис.14.12. Представлення невиявленої помилки в другому розряді кодів ПСК.

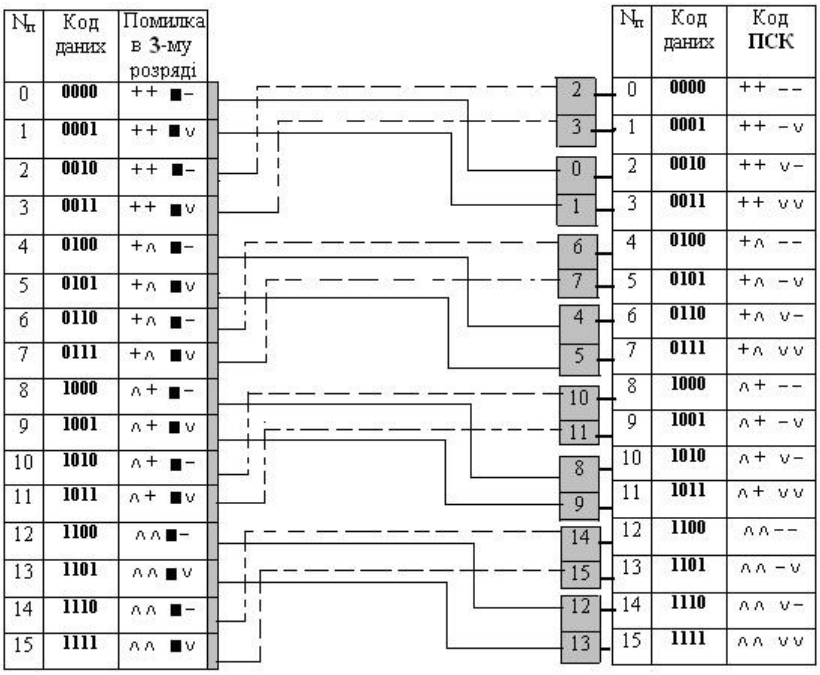


Рис.14.13. Представлення невиявленої помилки в третьому розряді кодів ПСК.

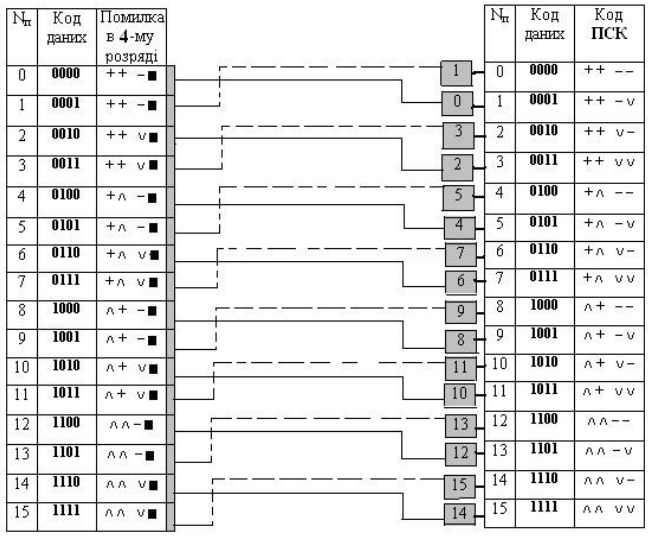


Рис.14.14. Представлення невиявленої помилки в четвертому розряді кодів ПСК.

Отже, із 16 –ти варіантів передавання даних півбайтовими кодами не виявляється 25% помилок в потоці даних.

Розглянемо приклад можливості виявлення помилок при формуванні ПСК даних довільної довжини $N = 2^k$, $k = 2,3,\ldots$, при використанні кодів Галуа типу G_2^2, G_2^3, G_2^4,…, G_2^k, згідно рекурентної процедури:

$$G_{i+i}=G_i \oplus G_{i-(n-1)}\,, \qquad (14.9)$$

де n - довжина ключа коду Галуа, $\oplus$ - операція додавання “по модулю два”.

$G_2^2 = 1100\ldots$

$G_2^3 = 11101000\ldots$

$G_2^4 = 1111010110010000\ldots$

$G_2^{10} = 11111111110101010110011\ldots$

Позитивною характеристикою ПСК є можливість виявлення та в окремих випадках виправлення помилок типу «стирання» та «вставок» бітів, які можуть виникати під впливом мультиплікативних завад. Функціональним обмеженням є відсутність можливості визначення числа нулів та одиниць в блоці даних.

В табл.(14.11) показано приклади формування ПСК для різних кодових комбінацій даних. Структура даної таблиці включає три групи даних.

В першій групі приведена реалізація символів сигнальних кодів для різних полів Галуа G_2^2, G_2^3, G_2^4, G_2^{10}. У випадку формування потоку нульових даних.

В другій групі приведений приклад реалізації потоку даних, який кодується ПСК в різних полях Галуа на основі чотирьох ознак маніпуляції $+,-,\wedge,\vee$.

В третій групі приведені потоки маніпульованих сигналів ПСК при передаванні потоку даних у вигляді послідовності одиниць.

Таблиця 14.11.

Формування ПСК при різних кодових комбінацій довжини даних.

	start	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24 ...	stop
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...	
G_2^2	2+1 ---	+	+	−	−	+	+	−	−	+	+	−	−	+	+	−	−	+	+	−	−	+	+	−	...	+++
G_2^3	3+1 ----	+	+	+	−	+	−	−	−	+	+	+	−	+	−	−	−	+	+	+	−	+	−	−	...	++++
G_2^4	4+1 -----	+	+	+	+	−	+	−	+	+	−	−	+	−	−	−	−	+	+	+	+	−	+	−	...	+++++
G_2^k	k=10 -----	+	+	+	+	+	+	+	+	+	+	−	+	−	+	−	+	−	+	−	−	+	+	+	...	+++++ ...
		1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1	...	
G_2^2	2+1 ---	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	+	$\wedge$	−	$\vee$	+	+	$\vee$	$\vee$	$\wedge$	+	−	−	$\wedge$	$\wedge$	$\vee$	...	+++
G_2^3	3+1 ----	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\vee$	$\vee$	+	$\wedge$	+	$\vee$	+	−	$\vee$	$\vee$	$\wedge$	+	+	−	$\wedge$	$\vee$	$\vee$	...	++++
G_2^4	4+1 -----	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	+	−	−	$\wedge$	−	−	$\vee$	$\vee$	$\wedge$	+	+	+	$\vee$	$\wedge$	$\vee$	...	+++++
G_2^k	k=10 -----	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	+	$\wedge$	−	$\wedge$	−	+	$\vee$	$\wedge$	$\vee$	+	−	+	$\wedge$	$\vee$	$\vee$	...	+++++ ...
		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...	
G_2^2	2+1 ---	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\vee$	...	+++
G_2^3	3+1 ----	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\vee$	...	++++
G_2^4	4+1 -----	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\vee$	$\vee$	$\vee$	$\vee$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	...	+++++
G_2^k	k=10 -----	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge$	$\wedge$	$\vee$	$\vee$	...	+++++ ...

Таким чином, при формуванні інформаційних повідомлень на основі ПСК незалежно від довжини об’єму масиву даних та різних модулюючих кодів Галуа ймовірність виявлення помилок постійна і число невиявлених помилок становить 25%.

В (табл.14.12) показано приклади невиявлених помилок в ПСК при різних модулючючих кодах Галуа: G_2^3, G_2^4, з якої видно, що число невиявлених помилок також становить 25%.

Таблиця 14.12.

Коректуючі властивості ПСК при різних кодових комбінаціях довжини даних.

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
Д	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G_2^3	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0
	+	+	+	—	+	—	—	—	+	+	+	—	+	—	—	—
+ | < >	■	■	■	^	■	^	^	^	■	■	■	^	■	^	^	^
	v	v	v	■	v	■	■	■	v	v	v	■	v	■	■	■
	—	—	—	+	—	+	+	+	—	—	—	+	—	+	+	+
G_2^4	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
	+	+	+	+	—	+	—	+	+	—	—	+	—	—	—	—
+ | < >	■	■	■	■	^	■	^	■	■	^	^	■	^	^	^	^
	v	v	v	v	■	v	■	v	v	■	■	v	■	■	■	■
	—	—	—	—	+	—	+	—	—	+	+	—	+	+	+	+
Д	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
G_2^3	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0
	^	^	^	v	^	v	v	v	+	+	+	—	+	—	—	—
+ | < >	v	v	v	■	v	■	■	■	■	■	■	^	■	^	^	^
	■	■	■	+	■	+	+	+	v	v	v	■	v	■	■	■
	—	—	—	—	—	—	—	—	—	—	—	+	—	+	+	+
G_2^4	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
	^	^	^	^	v	^	v	^	+	—	—	+	—	—	—	—
+ | < >	v	v	v	v	■	v	■	v	■	^	^	■	^	^	^	^
	■	■	■	■	+	■	+	■	v	■	■	v	■	■	■	■
	—	—	—	—	—	—	—	—	—	+	+	—	+	+	+	+
Д	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G_2^3	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0
	^	^	^	v	^	v	v	v	^	^	^	v	^	v	v	v
+ | < >	v	v	v	■	v	■	■	■	v	v	v	■	v	■	■	■
	■	■	■	+	■	+	+	+	■	■	■	+	■	+	+	+
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
G_2^4	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
	^	^	^	^	v	^	v	^	^	v	v	^	v	v	v	v
+ | < >	v	v	v	v	■	v	■	v	v	■	■	v	■	■	■	■
	■	■	■	■	+	■	+	■	■	+	+	■	+	+	+	+
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

14.4.2. Метод формування несиметричного рекурентного сигнального коду (НРСК).

Принцип формування НРСК полягає в тому, що послідовність нулів, які передаються в пакеті даних, нумерується рекурентним кодом Галуа G_2^k. Причому біт Галуа “1” передається фронтом спаду, тобто маніпуляційним

сигналом “10”, а нулі бітів Галуа передаються сигналом “00”. Для передавання одиниць використовується фронт наростання. В результаті такого способу формування сигналів виникає можливість виявлення помилок при передаванні даних, на базі рекурентних властивостей коду Галуа (рис.14.15, табл.14.13).

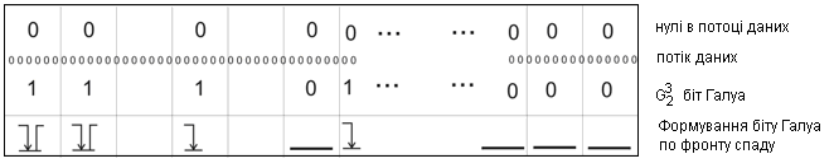
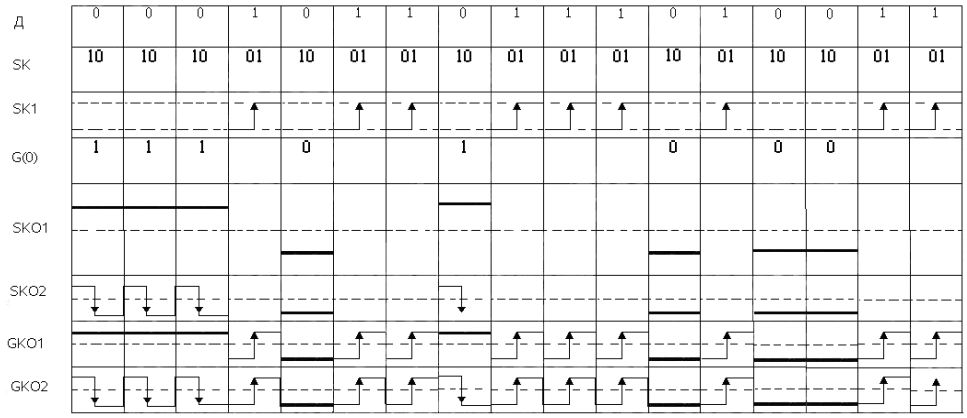


Рис.14.15. Метод формування нулів G_2^3 в потоці даних.

Таблиця 14.13.

Реалізація методу сигнального кодування двох байт даних , з використанням кодової послідовності Галуа G_2^3 (11101000).



В (табл.14.13) введені наступні позначення.:
Д- дані, які передаються; SK – сигнальний код каналу зв'язку (Манчестерський код); SK1–сигнальний код передавання "1" в каналах зв'язку; G(0) – код Галуа, який нумерує нулі; SK01 – сигнальний код нулів, коли біт- Галуа "1" передає кодом "11", а біт-Галуа "0" передає кодом "00"; SK02 – сигнальний код нулів, коли біт-Галуа "1" передає "10", а "0" —> "00"; GK01 – сигнальні коди маніпуляції, в яких нулі нумеруються кодом Галуа; GK02 – сигнальні коди маніпуляції, в яких нулі нумеруються кодом Галуа по фронту спаду.

Сигнальний код GK02, внаслідок використання сигналів фронту спаду для одиничних бітів коду Галуа, забезпечує кращу бітову синхронізацію по відношенню до сигнального коду GK01, в якому використовуються тільки потенціальні сигнали.

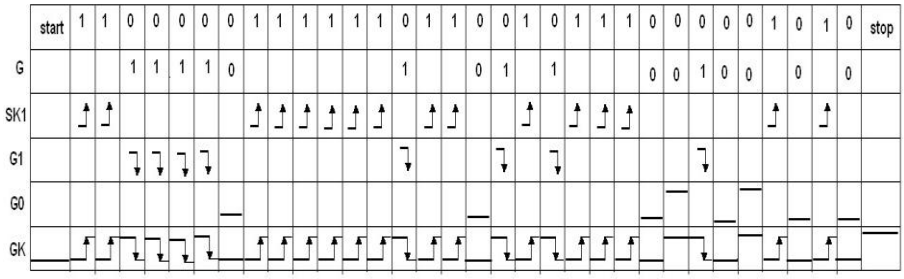
Функціональним обмеженням такого коду є відсутність можливості визначення загального числа нулів в блоці даних, оскільки модуляційний код Галуа повторюється багато разів.

Крім того, викладений принцип сигнального кодування даних в базисі Галуа, як видно з (табл.14.13), характеризується функціональним обмеженням, який полягає в тому, що при повторенні нулів і їх кодуванні в базисі Галуа потенціальними сигналами «-1» відсутня бітова синхронізація. Тому для реалізації ефективної бітової синхронізації запропоновано модифікацію несиметричних рекурентних кодів на основі використання зміни потенціалу «-1» на потенціал «+1» при однократному повторенні Галуа кодованих послідовностей нулів і кодуванню одиниць Галуа фронтом спаду.

В (табл.14.14) наведений приклад модифікованого методу безнадлишкового сигнального кодування з бітовою синхронізацією інформації в базисі Галуа на основі НРСК кодів, який характеризується ефективною блоковою і бітовою синхронізацією, можливістю виявлення та виправлення помилок типу зміни значення біту на протилежне, а також стирань та вставок бітів внаслідок кодових завад.

Таблиця 14.14.

Метод безнадлишкового кодування 4 байт даних в базисі Галуа на основі НРСК кодів з бітовою синхронізацією.



При застосуванні безнадлишкового методу сигнального формування даних з захистом від помилок на основі кодів поля Галуа структура фрейму має вигляд, представлений на (рис.14.16).

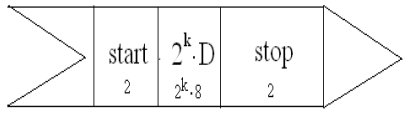


Рис.14.16. Структура фрейму безнадлишкового коду НРСК.

Згідно структури, оцінку надлишковості передавания даних розраховуємо за формулою:

$$K_{н\ (G1)} = (4 + 2^k \cdot 8) / K_{ФС} \cdot 2^k \cdot 8 , \qquad (14.10)$$

де $K_{ФС} = 0.75$.

Приклад реалізації коду НРСК в сигнальному просторі ±1 показано на (рис.14.17).

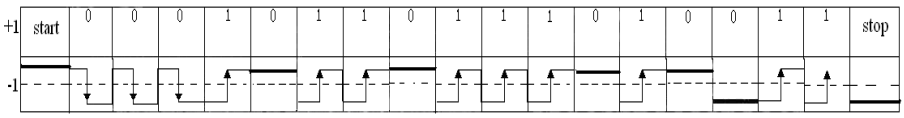


Рис.14.17. Структура НРСК в сигнальному просторі ±1.

Функціональними обмеженнями модифікованого НРСК є відсутність визначення числа нулів в блоці даних в тому випадку, коли довжина модуляційного коду Галуа є менша від довжини коду даних.

14.4.3. Формування рекурентного симетричного сигнального коду (РССК).

Другою модифікацією НРСК є використання коду Галуа для одночасного симетричного сигнального формування нулів і одиниць потоку даних, при чому об'єм коду Галуа відповідає об'єму даних, що передаються (РССК).

В табл.(14.15) показані приклади формування сигналів даної модифікації РССК, при G_2^4, де сигнали +1 та -1 нульових позицій коду Галуа формуються згідно нульових позицій коду Галуа.

Таблиця 14.15.

Формування сигналів модифікованим РССК, при G_2^4.

		d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
	Д	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	G_2^4	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
	SK	⤵	⤵	⤵	⤵	—	⤵	—	⤵	⤵	—	—	⤵	—	—	—	—
	CK	⌣	⌣	⌣	⌣	+	⌣	+	⌣	⌣	+	+	⌣	+	+	+	+
2	Д	1	1	1	1	0	0	0	0	1	0	1	1	0	0	1	1
	$1G_2^4$	1	1	1	1					0		1	0			1	1
	SK1	⤴	⤴	⤴	⤴					—		⤴	—			⤴	⤴
	$0G_2^4$					1	1	1	1		0			1	0		
	SK o					⤵	⤵	⤵	⤵		—			⤵	—		
	CK	⌢	⌢	⌢	⌢	⌣	⌣	⌣	⌣	—	+	⌢	—	⌣	+	⌢	⌢
3	Д	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	G_2^4	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
	SK	⤴	⤴	⤴	⤴	—	⤴	—	⤴	⤴	—	—	⤴	—	—	—	—
	CK	⌢	⌢	⌢	⌢	—	⌢	—	⌢	⌢	—	—	⌢	—	—	—	—

В (табл.14.15) SK і СК – відповідно сигнальні і символьні коди.

З (табл.14.15) видно, що в блоці даних об'ємом $N=2^4$ завершення послідовності нулів відповідає коду Галуа 1010 і завершується символами ∨+ ∨+, тобто N=7, згідно $G_2{}^4$.А завершення послідовності одиниць в коді Галуа відповідає символам ∧– ∧ ∧, тобто коду Галуа 1011, N=9.

Функціональним обмеженням лассу методу формування сигнального коду є недостатньо ефективна символьна (бітова) синхронізація при повторенні потенціальних сигналів +1 або -1.

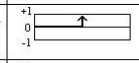
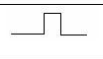
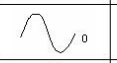
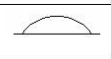
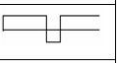
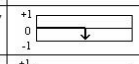
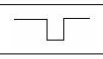
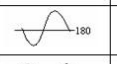
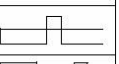
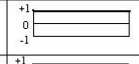
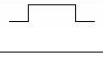
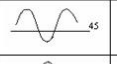
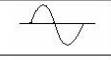
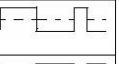
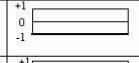
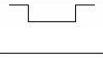
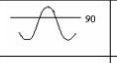
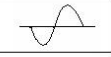
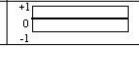
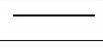
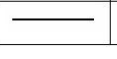
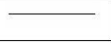
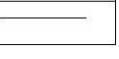
Таким чином, РССК забезпечує ефективне симетричне кодування у вигляді кодів Галуа послідовності нулів і одиниць блоку даних з однозначним визначенням їх числа $N_0 + N_1 = N$, яке може бути використане для виявлення та виправлення помилок після передавання даних в комп'ютерних системах.

14.4.4. Метод формування безнадлишкових квазісимвольних сигнальних кодів (КССК).

Якщо розглянути сигнальне вікно маніпуляції двійкової інформації для ансамблів амплітудних, імпульсних, частотних, фазових та кодових сигналів 2^k+1, k=2, то можна визначити п'ять ознак маніпуляції наступного виду (табл.14.16).

Таблиця 14.16.

Ознаки маніпуляції ансамблів сигналів.

	ОЗНАКИ МАНІПУЛЯЦІЇ СИГНАЛІВ				
	Потенціальні	Імпульсні	Фазові	Частотні	Кодові
∧					
∨					
+					
–					
S					

Будь-яка з п'яти ознак може бути використана в якості синхросигналу (S), який використовується при повторенні однієї з інших ознак сигналів, тому дані методи маніпуляції належать до лассу квазісигнальних. Найпростішим прикладом такого лассу сигналів є квазітрійковий код КТ-1 (табл.14.5).

Такі способи маніпуляції відповідають числу сигнальних ознак $N=2^k+1$, де k=2, а S=0 потенціал «—», причому потенціал нуль використовується для виключення повторень інших сигналів, що забезпечує якісну символьну синхронізацію на основі сигнальних просторів .

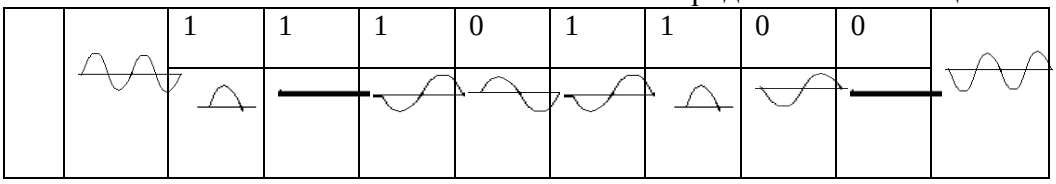
На основі приведених в табл.14.16 способів маніпуляції сигналів можлива відповідна розробка методу квазісимвольного способу формування СК на основі кодів Галуа. Приклад такого формування СК для однобайтових даних показано в табл.14.17.

Таблиця 14.17.

Квазісимвольний СК з кодом Галуа “1100”.

	Start	x	x	x	x	x	x	x	x	Stop
Потенціальні сигнали	S	0	0	0	0	0	0	0	0	
	S	+	S	–	S	+	S	–	S	– –
	S	0	1	0	1	0	1	0	1	
	S	+	∧	+	∧	–	∨	–	∨	– –
	S	0	0	0	0	1	1	1	1	
	S	+	S	–	S	∧	S	∨	S	– –
	S	0	0	1	1	0	0	1	1	
	S	+	S	∧	S	–	S	∨	S	– –
	S	0	0	0	1	1	1	0	0	
	S	+	S	–	∧	S	∨	–	+	– –
	S	1	1	0	0	0	1	0	1	
	S	∧	S	+	S	–	∨	–	∨	– –
	S	1	0	0	0	0	0	1	1	
	S	∧	+	S	–	S	+	∧	S	– –
Фазові сигнали		0	0	0	0	0	0	0	0	
		0	1	1	0	0	1	1	0	
		1	1	1	0	0	1	1	0	
Част.		1	0	0	0	0	1	1	1	

продовження таблиці 14.17.



Отже, для будь –якої довжини біт-орієнтованого блоку даних об'ємом $M=2^k$ можна вибрати код Галуа БСК з об'ємом $N=M/2^k$, де k = 0,1,2,…n.

Наприклад, для байт-орієнтваних блоків даних ($M=2^3$ = 8 біт) можна вибрати код Галуа 1100 з двохбітовим ключем (табл.14.17), або код Галуа 1110100 з трьохбітовим ключем.

Тоді отримаємо наступні послідовності симетричного СК, згідно (табл.14.18), для ансамблю потенціальних сигналів.

Таблиця 14.18.

Симетричний квазісимвольний рекурентний код.

	Код Галуа	start	x	x	x	x	x	x	x	x	stop
A	1110100	∨ ∨ ∨	**0** +	**0** S	**0** +	**0** –	**1** ∧	**1** S	**1** ∧	**1** ∨	∧ ∧ ∧
B	1110100	∨ ∨ ∨	**1** ∧	**0** +	**0** S	**1** ∧	**1** ∧	**1** ∨	**0** +	**1** ∧	∧ ∧ ∧
C	1110100	∨ ∨ ∨	**1** ∧	**1** ∧	**1** ∧	**1** ∨	**1** ∧	**1** ∨	**1** ∨	**1** ∧	∧ ∧ ∧

Підвищення якості синхронізації квазісимвольного симетричного коректуючого коду, згідно табл.2.21 показано на рис.14.18.

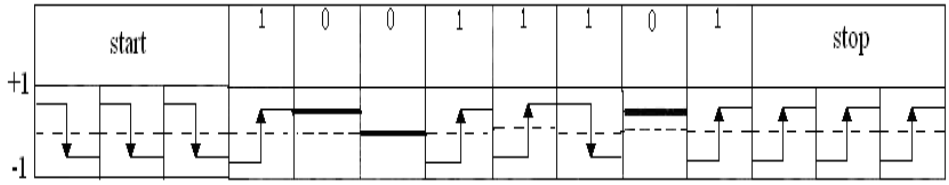


Рис.14.18 Реалізація квазісимвольного симетричного рекурентного коректуючого коду.

В сучасних комп'ютерних системах та мережах використовуються стандартні пакети передачі повідомлень, (табл.14.19).

Таблиця 14.19.

Структура фрейму мережі Ethernet.

Преамбула	Адреса одержувача	Адреса відправник	Тип кадру	Дані	ЦКС (crc)
64 біта	48 біт	48 біт	16 біт	Від 512 до 32 000 біт	32 біт

Аналіз структури кадру Ethernet (табл.14.19) показує, що згідно формули (2.6) надлишковість змінюється в границях від 1.40 до 1.008 при зміні об'єму даних, що передаються, від 512 до 32 000 біт:

$К_{н(Ethernet)}$ = 1.406, при 512 ; $К_{н(Ethernet)}$ = 1.008, при 32 000. Причому надлишковість циклічного контрольного коду (crc), який виявляє помилки, відповідно змінюється в границях від 1.00625 до 1.

В середньому надлишковість передавання даних, за рахунок виникнення помилок в інформаційних каналах та повторних передавань пакетів, в залежності від типу каналів зв'язку, становить від 10 до 30 %. Це відповідає зростанню надлишковості кодування даних в мережі Ethernet, яку можна оцінити згідно виразу:

$$К_н = (n+m)/n \cdot К_{пд}, \qquad (14.11)$$

де $К_{пд}$ – коефіцієнт повторних передач.

Таким чином, оцінка надлишковості передавання даних в мережі Ethernet, з врахуванням повторних передач даних, що передаються в каналах зв'язку, відповідно змінюється в границях від 1.82 до 1.3.

Отже, встановлена однозначність безнадлишкового сигнального кодування інформаційних потоків даних з використанням типових ансамблів потенціальних та інших сигналів з можливістю виявлення та виправлення помилок кодами Галуа дозволяє суттєво зменшити надлишковість існуючих методів кодування даних, що використовуються в комп'ютерних мережах. При цьому є доцільним поглиблене дослідження коректуючих властивостей запропонованих методів сигнального кодування на основі кодів Галуа.

14.5. Виявлення та виправлення помилок при використанні сигнальних кодів Галуа.

Виявлення та виправлення помилок на основі запропонованого методу та способів сигнального кодування повідомлень реалізується на приймальному кінці ІК, шляхом використання рекурентних властивостей кодів поля Галуа, які додатково модулюють біт-орієнтовані дані, що передаються.

У випадку, якщо код "1" в даних прийнято неправильно і замість "1" декодер формує нуль, це означає, що в середовищі даних міститься

дев'ять нулів (в той же час код "00" в кінці передачі містить 8 нулів). Тобто при появі будь-якого числа помилок в переданих даних кінцевий код Галуа нулів не буде відповідати числу прийнятих, що дозволяє виявити помилку.

Розглянемо приклад ідентифікації помилок в масиві даних, що передаються, в табл.14.20.

Таблиця 14.20.

Реалізація потоку даних, що кодується НРСК кодом, з виявленням помилок на сигнальному рівні.

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Д	1	1 0	2 0	1	3 0	1	4 0	1	1	1	5 0	1	1	6 0	1	1	7 0	1	1	1	1	8 0	1	0	1 …
G(0)		1	1		1		0				1			0			0					0		1	
СК(0)		↴	↴		↴		—				↴			—			—					—		↴	
СК(1)	↥	.	.	↥	.	↥	.	↥	↥	↥	.	↥	↥	.	↥	↥	.	↥	↥	↥	↥	.	↥	↥	
*							*			*							*				*				
SK *	∧	∨	∨	∧	∨	∧	◇ +	∧	∧	∨ +	∨	∧	∧	∨	∧	∧	◇ +	∧	∧	∧	∨ +	∨	∧	∨	

* - сигнальний синдром помилки.

В табл.14.20 приведено приклад виникнення помилок на сигнальному рівні в 7-ій та 17-ій позиції нулів, а також 10-ій та 21-ій позиції одиниць.

Як показано в даній таблиці, помилки, що виникають на сигнальному рівні, при передаванні нулів, промодульованих кодом Галуа, однозначно виявляються, оскільки порушується рекурентність бітів Галуа. Крім того, порушується число фактичних символів нулів по відношенню до їх кількості, представлених кодом Галуа. Тобто $N_0^* = 6$, $N_0 = 8$, $N_0^* < N_0$, при будь-якій кількості спотворених помилками нульових бітів Галуа.

При спотворенні помилками на сигнальному рівні бітів даних одиниць, як це показано в табл.14.20, можливе виникнення наступних варіантів декодування сигнального коду:

1) фронтом наростання (∧) символу біта даних одиниці на сигнальному рівні представляється потенціалом “+”, що відповідає поняттю стирання даного біта, і ця помилка однозначно виявляється декодером.

2) фронтом спаду (∨), або потенціалом “–“ , що відповідає на рівні декодера Галуа-одиниці або Галуа-нулю, нульових бітів даних. При цьому однозначно порушується рекурентна послідовність бітів Галуа, а фактичне число прийнятих бітів нулів $N_0^* = N_0+-1$, що приводить до порушення рекурентності бітів Галуа.

Дослідимо можливість виявлення та виправлення помилок в квазісигнальному рекурентному коді.

Приклад потоку даних з позиціями появи помилок показано в табл.14.21.

Таблиця 14.21.

Реалізація потоку даних, що кодується КССК з виявленням помилок на сигнальному рівні.

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Д G_1 G_0	1 1	1 1	1 1	1 1	0 1	0 1	0 1	0 1	0 0	0 1	0 1	0 0	1 1	0 1	0 1	0 1	1 1	0 0	0 0	1 1	0 1	0 1	0 0	0 1	1 …
KSCK	⤴	⤴	⤴	⤴	—	—	—	—	_	—	⤵	_	⤴	—	_	⤵	⤴	_	—	⤴	⤵	⤵	_	—	
КССК	^	^	^	^	+	S	+	S	−	+	v	−	^	+	S	v	^	−	S	^	v	v	−	+	
*				*			*					*				*					*				
СП*				v ± S			^ v − S					^ v + S				^ ± S					v ± S				

СП* - синдром виникнення помилки.

При виникненні помилок на сигнальному рівні і в одиницях потоку даних, можливі два випадки:

1) інвертування Галуа ознаки одиничного біта, що однозначно виявляється рекурентним декодером потоку Галуа-одиниць;

2) заміна сигнальної ознаки одиниць, які представляються фронтом наростання та спаду, і перетворення їх в сигнальні ознаки нулів, які представляються потенціалами “+” , “–“ та “S-нуль“. Це призведе до стирання одиниці в даній позиції, що виявляється рекурентним декодером, та одночасно біт-стаффінгом нуля з ознаками Галуа одиниці або нуля чи “S“, що також виправляється рекурентним шляхом на рівні декодера.

Аналогічні сигнальні переходи з однозначним виявленням та виправленням одиничних помилок ідентифікуються на сигнальному рівні, при появі помилок в нульових бітах потоку даних.

Можливість виявлення помилок, при функціонуванні запропонованих методів кодування інформації в оптичних каналах зв’язку може бути реалізована у двох випадках:

1) виявлення помилок на приймальному кінці каналів зв'язку, ґрунтується на біт-орієнтованій нумерації послідовності нулів, які передаються кодовою послідовністю Галуа;

2) якщо помилка виявлена, використовуємо формулу (14.9) , де рекурентним шляхом можна перевірити, в якій саме позиції відбулася заміна символу нуля в процесі передавання даних.

Проведені дослідження методики сигнального кодування даних на основі коду Галуа у відкритих оптичних ІК показують, що оптимізація та

покращення методів цифрового опрацювання даних з виявленням та виправленням помилок ефективно реалізовуються на основі запропонованого безнадлишкового кодування даних та використанні сучасних методів цифрового оброблення сигналів.

14.6. Критерії оцінки і порівняльна характеристика ефективності виявлення та виправлення помилок сигнальними кодами.

Дослідження ефективності запропонованих методів формування СК оцінимо на основі наступної моделі.

Нехай передається блок даних об'ємом Д зі швидкістю передавання V_0, і надлишковістю коректуючого коду N біт, причому в кожному блоці даних виявляється помилка декодером цифрового приймача даних (рис.14.19).

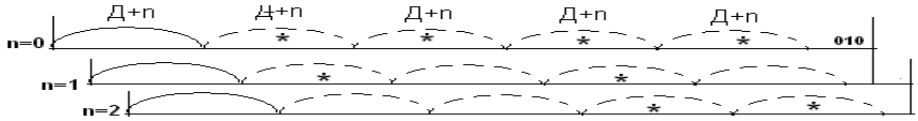


Рис.14.19. Модель передавання блоку даних з однократними помилками.

При цьому реальна швидкість передавання даних рівна нулю. Можливі наступні випадки, коли повторно передається кожний другий, четвертий, і наступні блоки даних. Тобто реальна швидкість передавання даних описується виразом:

$$V_X = V_0 \cdot (1 - 1/2^n \cdot k_n), \qquad (14.12)$$

де V_0 – базова швидкість передавання даних, n - кратність повторного передавання блоків даних, k_n - коефіцієнт надлишковості коректуючого коду.

На рис.14.20 показано графік зміни реальної швидкості передавання даних в комп'ютерній мережі, в залежності від числа повторної передачі блоків даних, без врахування k_n.

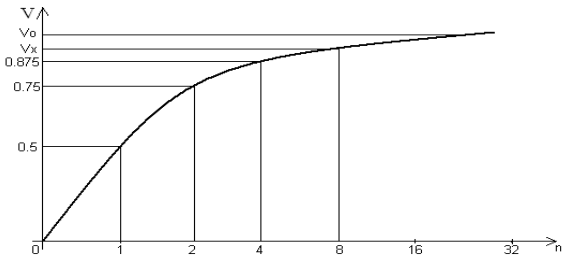


Рис.14.20. Залежність швидкості передавання даних від числа повторної передачі блоків даних.

При застосуванні запропонованих СК надлишковість кодування блоків даних для захисту від помилок відсутня, тобто $k_n = 1$.

За рахунок виявлення та виправлення будь-яких однократних помилок в блоках даних реальна швидкість передавання даних становить: $V_X^* = V_0 = \text{const}$, що відповідає відсутності повторної передачі даних. При цьому коефіцієнт ефективності передавання даних при застосуванні СК в базисі Галуа може бути розраховано згідно виразу:

$$k_{еф} = V_0^* - V_X,$$

$$\text{або } k_{еф} = V_0 - V_0 + V_0/ 2^n \cdot k_n = V_0/ 2^n \cdot k_n. \quad (14.13)$$

На рис.14.21. показано графіки коефіцієнтів СК в залежності від параметрів n та k_n.

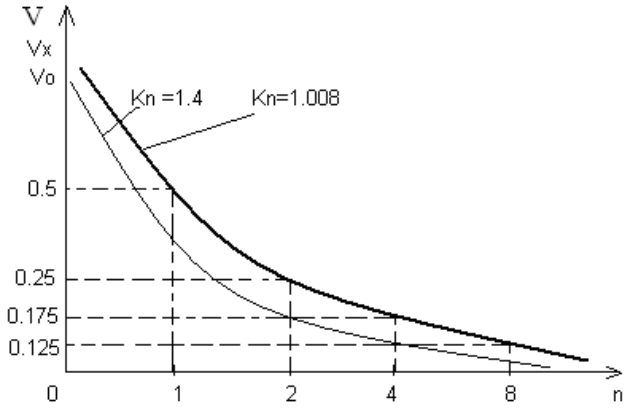
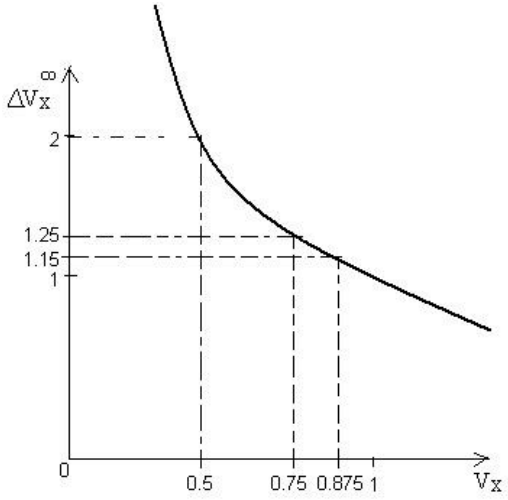


Рис.14.21. Графік залежності СК від параметрів n та k_n.

Для оцінки ефективності підвищення швидкості передавання даних на основі запропонованих СК доцільно використати критерій у вигляді співвідношення базової швидкості передавання даних, яку забезпечує запропонований метод, до реальної швидкості передавання даних, що досягається в існуючих комп'ютерних системах:

$$\Delta V_X^* = V_0^*/ V_X, \quad (14.15)$$

На рис.14.21 показано табличні та графічні результати оцінки ефективності підвищення швидкості передавання даних, на основі запропонованих СК по відношенню до існуючих , що використовуються в сучасних комп'ютерних системах.



Vx	ΔVx
0	∞
0.5	2
0.75	1.25
0.875	1.15

Рис.2.21. Результати оцінки ефективності підвищення швидкості передавання даних сигнальними коректуючими кодами.

Таким чином, встановлено, що запропонований метод побудови квазісимвольного СК забезпечує наступні позитивні характеристики:

1) використання характеристик сигнального вікна та потенціальнах кодів;

2) безнадлишковість розроблених сигнальних кодів, що дозволяють виявляти та виправляти однократні помилки;

3) обґрунтована перспективність застосування сигнальних кодів в існуючих комп'ютерних системах, а також в ІК комп'ютерних систем з відкритими оптичними каналами.

РОЗДІЛ 15

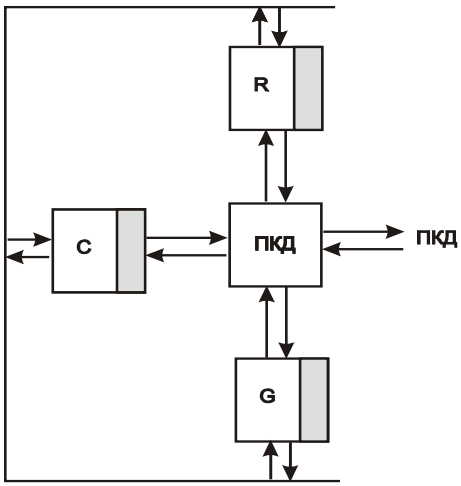
ПРОЦЕСОРИ ТА ЇХ КОМПОНЕНТИ В КОДАХ ПОЛЯ ГАЛУА ТА ГАЛУА-КРЕСТЕНСОНА

15.1. Формалізація операцій компонентів мультиядрового RCG процесора у базисі Крестенсона-Галуа.

Основним принципом створення RCG – процесора є одночасне використання теоретико-числових базисів Радемахера, Крестенсона, Галуа та управління найефективнішим перерозподілом виконання операцій на їх основі.

Світовий досвід проектування високопродуктивних процесорів свідчить, що найбільш ефективною архітектурою багатоядерного процесора є зірково-магістральна топологія. При цьому найбільш ефективними ТЧБ для реалізації такого класу процесорів є базиси Радемахера, Крестенсона та Галуа.

На рис. 15.1 показана структура мультиядрового процесора, яка реалізована згідно зірково-магістральної архітектури.



R – процесорний елемент в базисі Радемахера,
C – процесорний елемент в базисі Крестенсона,
G – процесорний елемент в базисі Галуа,
ПКД – пам’ять колективного доступу.

Рис.15.1. Структурана схема RCG процесора**.**

З рис.15.1 видно, що для ефективної координації роботи модулів та забезпечення максимальної продуктивності такого процесора необхідно використовувати пам’ять колективного доступу ПКД, яка б забезпечувала високий рівень паралелізму інтерфейсних зв’язків та дозволяла найбільш доцільно розподіляти ресурси процесорів, реалізованих у різних ТЧБ.

Блоки виконання математичних операцій в різних теоретико-числових базисах представляють собою окремі незалежні один від одного

процесорні елементи, що дозволяють здійснювати паралельно-послідовну обробку даних, згідно заданих алгоритмів, що формуються контролером команд.

В якості блоку виконання математичних операцій в базисі Радемахера використовується ядро класичного швидкодіючого арифметико-логічного пристрою в двійковій системі числення.

Запропоновано принцип спільного використання кодового базису Крестенсона та Галуа при реалізації блоку виконання математичних операцій реалізовано у базисі Крестенсона, в якому найефективніше виконуються операції модульного додавання та множення, згідно якого двійкове представлення залишків здійснюється не в двійковій системі числення, а в рекурентних послідовностях Галуа (рис. 15.2).

a0	M0
a1	M1
a2	M2
a3	M3
a4	M4
a5	M5
a6	M6

а)

a0	a1	a2	M0
a1	a2	a3	M1
a2	a3	a4	M2
a3	a4	a5	M3
a4	a5	a6	M4
a5	a6	a0	M5
a6	a0	a1	M6

б)

1	1	1	0
1	1	0	1
1	0	1	2
0	1	0	3
1	0	0	4
0	0	1	5
0	1	1	6

в)

Рис 15.2. Представлення діапазону чисел по модулю 7 в кодах Галуа. аі – біти Галуа, Мі – коди залишків по модулю 7, а) – таблиця відповідності бітів Галуа до чисел множини М, б) – рекурентне представлення залишків бітами Галуа, в) – кодова матриця представлення залишків кодами Галуа.

Рис. 15.2 показує реалізацію однозначного представлення кодів залишків базису Крестенсона в базисі Радемахера відповідними бітами та рекурентними послідовностями кодів Галуа. При цьому необхідно виконувати наступні умови:

$$G\begin{pmatrix} b \\ a \end{pmatrix};$$

$$G\begin{pmatrix} n \\ 2 \end{pmatrix},$$

$$N_i = 2^n - 1,$$

$$P_i = N_i \rightarrow P_i = 2^n - 1,$$

для $G\begin{pmatrix} b \\ a \end{pmatrix}$, $N_i = 2^n - 1$, причому $P_i = N_i$.

$$\left.\begin{array}{l} N\left[G\begin{pmatrix} n \\ 2 \end{pmatrix}\right] = 2^n - 1; \\ P_i = N\left[G\begin{pmatrix} n \\ 2 \end{pmatrix}\right]; \end{array}\right\}, \qquad (15.1)$$

Звідки $P_i=2^n-1$.

З виразу (15.1) випливає, що P_i базису Крестенсона одночасно відповідають модулям кілець базису Галуа, тобто (3, 5, 7, 15, 31, і т.д.).

Така реалізація способу кодування інформації у базисі Крестенсона-Галуа забезпечує можливість одночасного використання переваг рекурентних кодів поля Галуа та модульних операцій базису Крестенсона.

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

Інтегрально-імпульсний перетворювач (ІІП) призначений для опрацювання та передавання даних про стан технологічних об'єктів. ІІП є важливим компонентом на низовому рівні комп'ютерної мережної системи контролю та обліку витрати енергоносіїв Alfiya.

ІІП складається з перетворювача напруга – частота (ПНЧ), масштабуючого подільника частоти (МД), кодер-інтегратора Галуа (KIG) і широтного модулятора (ШІМ) (рис. 15.3).

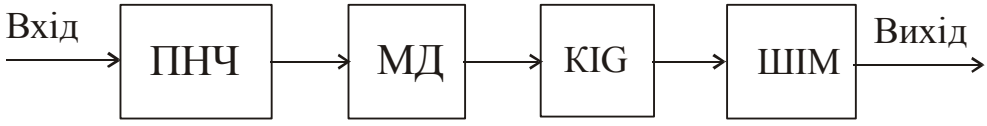


Рис. 15.3. Структурна схема ІІП.

Досвід проектування та експлуатації вказаного типу ІІП показав ряд функціональних та конструктивних особливостей:

1) функціональні особливості:

– жорсткий ключ для формування кодової послідовності Галуа у полі $GF(\begin{smallmatrix} 20 \\ 2 \end{smallmatrix})$ у діапазоні чисел $0 \le N < 2^{20} - 1$;

– наявність каналового входу;

– фіксована тривалость імпульсів 0 і 1 на виході ІІП;

– використання кристалу ПНЧ з двополярним живленням;

2) конструктивні особливості:

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

На рис.15.4 показана структура ІІП з розширеними функціональними можливостями на мікро-елементній базі розроблена автором сумісно з О.М. Заставним.

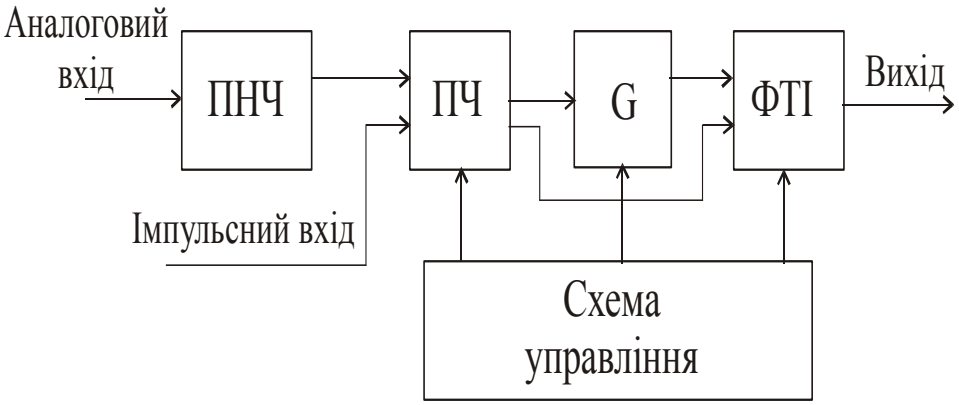


Рис. 15.4. Структурна схема вдосконаленого ІІП:
ПЧ – подільник частоти;
G – генератор коду Галуа;
ФТІ – формувач тривалості імпульсів.

Розроблений ІІП забезпечує наступні функціональні параметри (рис.15.5):

– 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.

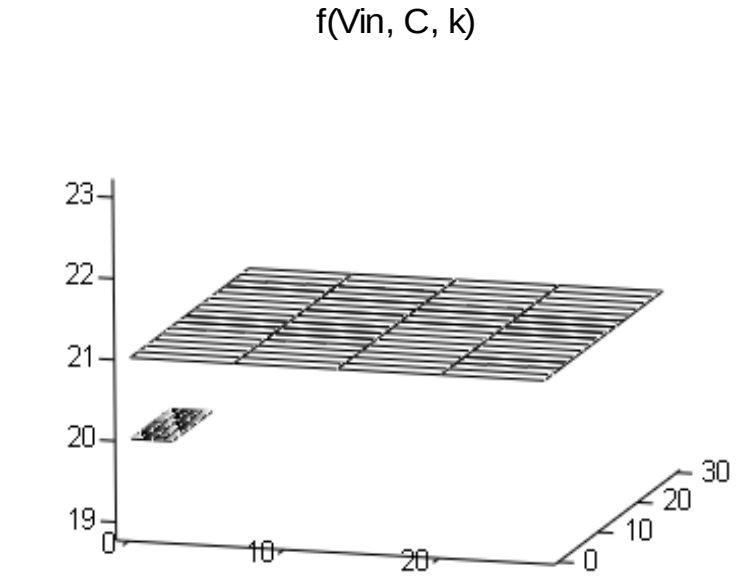


Рис. 15.5. Порівняння характеристик базовогоі вдосконаленого ІІП:
Vin – діапазон зміни вхідної напруги;
C – вибір тривалості вихідних імпульсів;
k – вибір ключів.

При розробці ІІП 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 ($50 \cdot 10^{-6} / ^{\circ}C$), 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 ІІП реалізовані на програмованій логічній інтегральній схемі (ПЛІС) фірми ALTERA EPM3064ALC44 (рис. 15.6):

– 8-розрядний двійковий лічильник count_8;
– мультиплексор mul_8_1;
– генератора коду Галуа g_generat;
– формувач тривалості імпульсів – fti.

Виміряний 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.

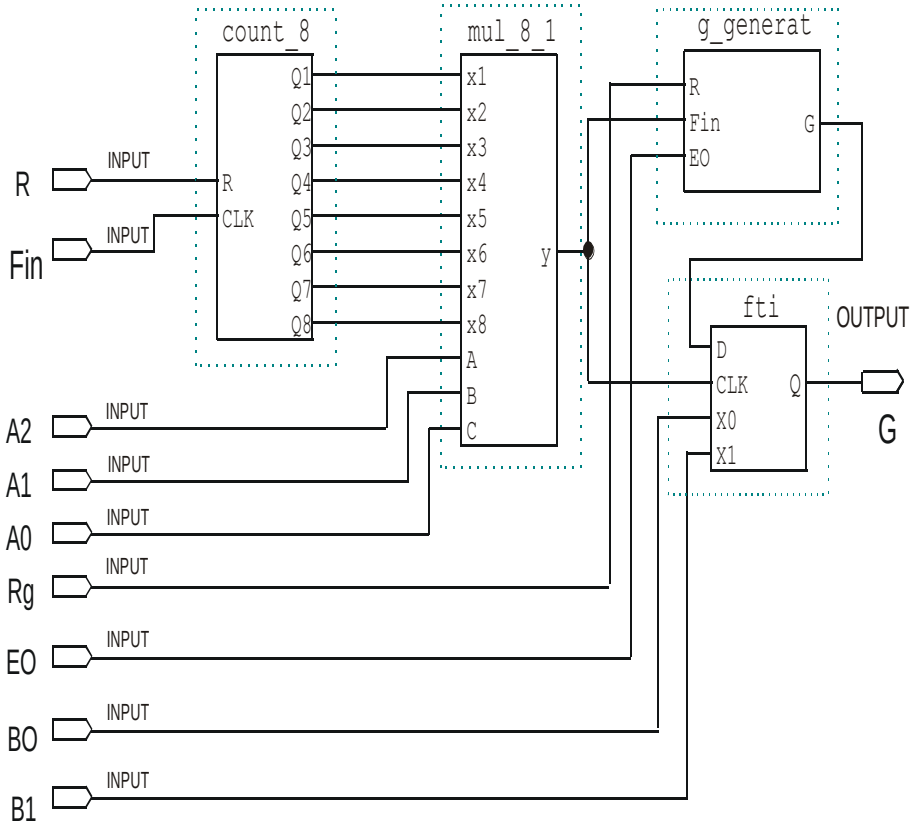


Рис. 15.6. Функціональна схема ІІП.

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. 15.1. Процедура генерування кода Галуа написана на AHDL у виді параметризованої макрофункції, яка описує пристрій, спрощена структурна схема якого приведена на рис. 15.7.

Таблиця 15.1.

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 Е. Результати моделювання роботи ІІП представлені на рис. 15.7.

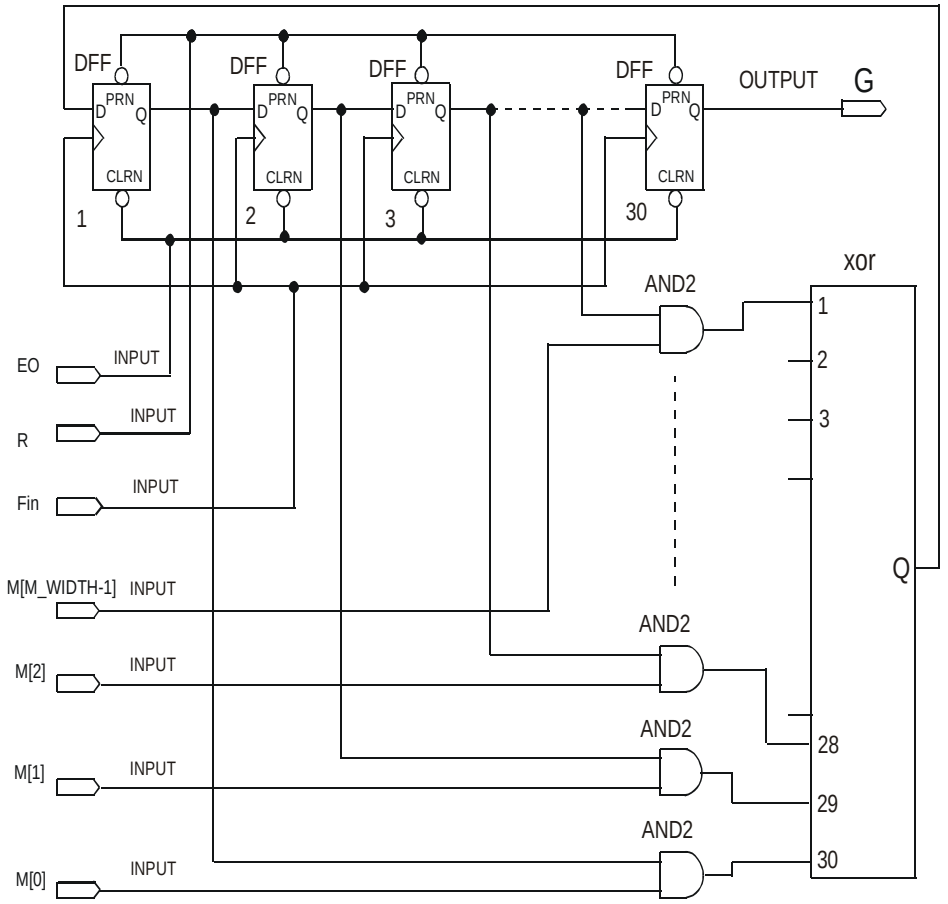


Рис. 15.7. Структурна схема блоку G_GENERAT.

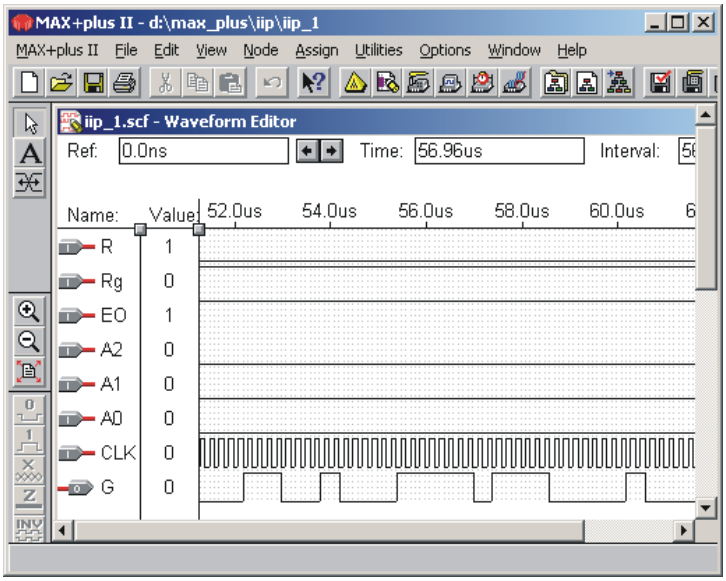


Рис. 15.8. Результати моделювання роботи ІІП.

Розробку ІІП виконано з використанням САПР MAX+PLUS II, яке дозволяє на рівні часових діаграм оцінити поведінку пристрою до програмування його в ПЛІС (рис.15.8). 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. 15.9).

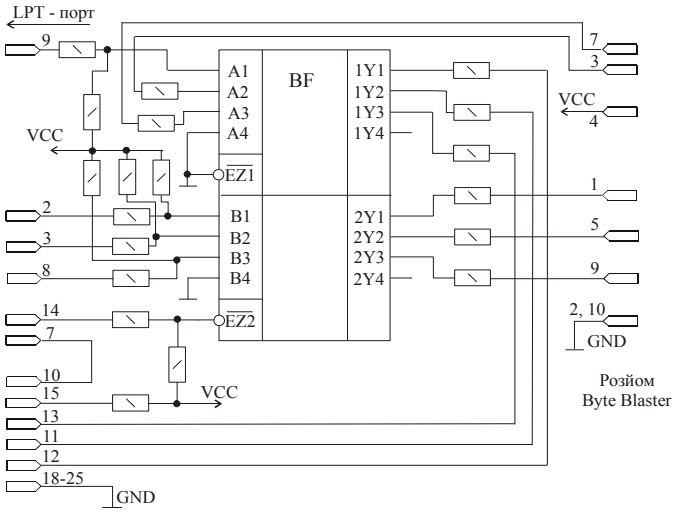


Рис. 15.9. Принципова схема завантажувального кабелю Byte Blaster MV.

Призначення контактів розніму Byte Blaster MV в режимі програмування приведено в табл. 15.2 .

Живлення завантажувального кабелю Byte Blaster MV здійснюється від джерела живлення І І П .

Таблиця 15.2.

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.15.10) дозволила підвищити надійність пристрою, суттєво прискорити час проектування та виготовлення, а отже, знизити вартість пристрою.

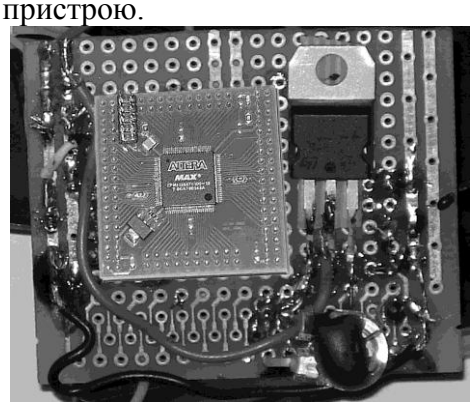


Рис. 15.10. Дослідний взірець ІІП.

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

15.3. Спецпроцесор опрацювання 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 передавання даних C_i, то, як правило

$$C_{K.З.} = \sum_{n=1}^{i} C_i .$$

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 (рис. 15.11) і призначений для кодування та передавання даних багатовимірних джерел інформації розподілених комп'ютерних систем контролю та керування технологічними процесами

Перетворення багатовимірного вектора в одновимірний відбувається за наступним алгоритмом: унітарний код , що відповідає значенню технологічного параметру поступає на вхід додавання UP лічильників Counter_P1, Counter_P2, Counter_P3, які працюють по модулях $P_1, P_2, \ldots P_n$ (рис. 15.7).

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:

$$N_i = \sum_{i=1}^{n} b_i \cdot B_i \ (\mathrm{mod}\ \wp),$$

де b_i – залишки, що відповідають значенню технологічних параметрів;

B_i – базисні числа;

$\wp = P_i \cdot P_2 \cdot \ldots \cdot P_n$ – модулі СЗК.

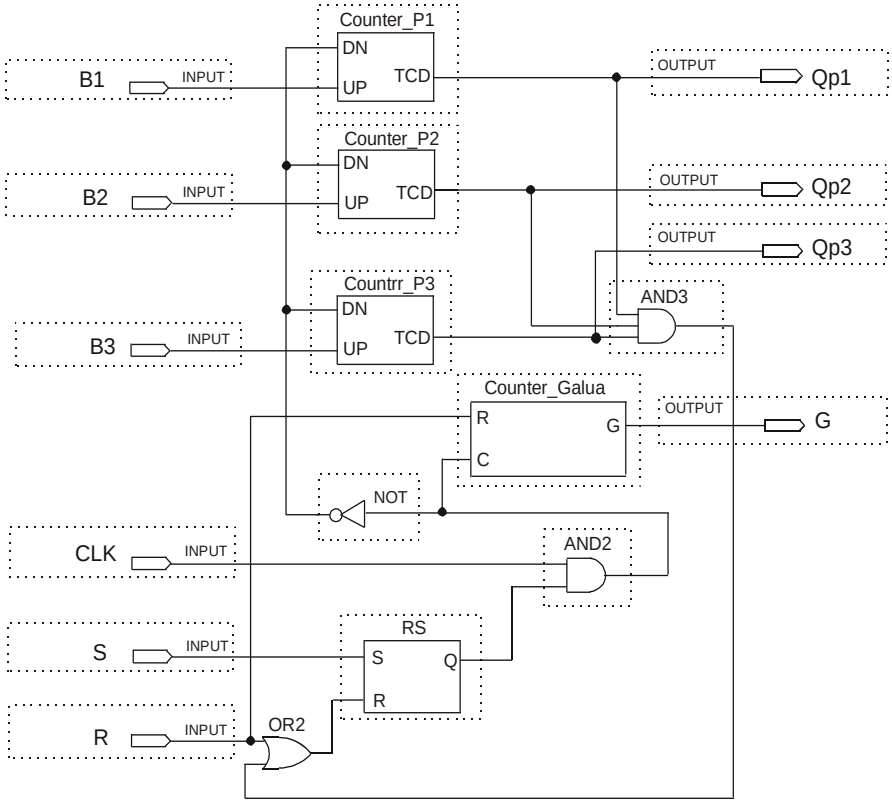


Рис. 15.11. Функціональна схема кодера.

В канал зв'язку передається останніх n – біт послідовності Галуа. На приймальній стороні код Галуа надходить на декодер Галуа (рис. 15.12), на виході якого отримуємо унітарний код, що поступає на лічильники, які ведуть підрахунок по модулях $P_1, P_2, \ldots P_n$. 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 І, буде логічна одиниця, яка встановить RS – тригер в положення "0".

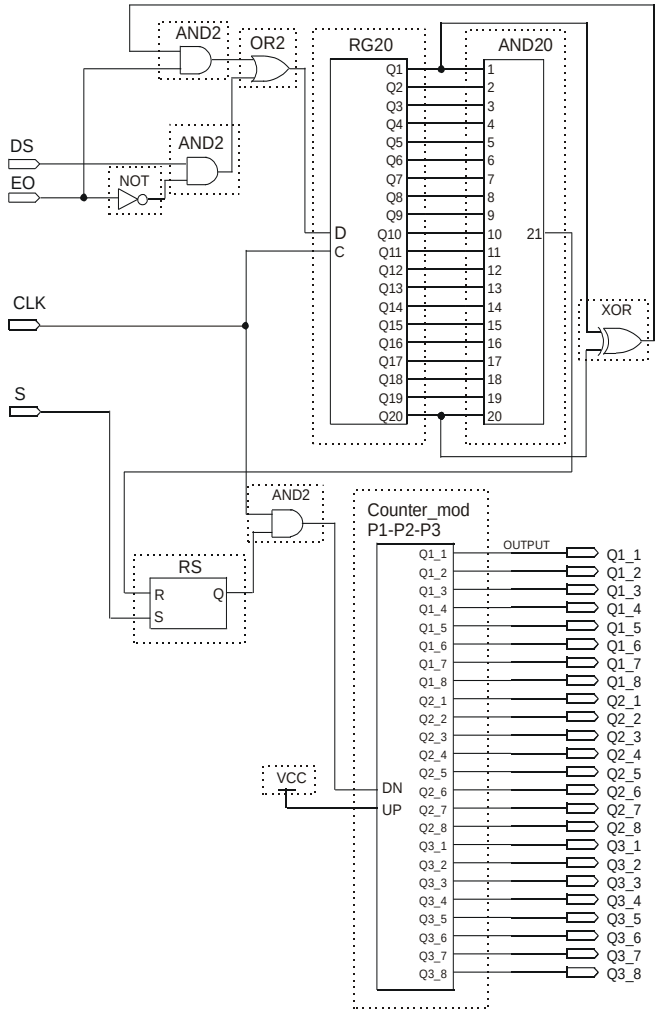


Рис. 15.12. Функціональна схема декодера.

В лічильнику CT_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 виміряним значенням технологічних параметрів.

Використання прямого перетворення СЗК дозволяє перейти від багатовимірного представлення виміряних 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.

15.4. Процесор стиснення даних на основі базисних функцій Галуа.

Процесор призначений для реалізації методу стиснення даних на основі базисних функцій Галуа нульового то першого порядку 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 15.13) складається з блока ідентифікації стану (БІ), дешифратора (DC), генератора Галуа, цифрового компаратора (ЦК) і ключа (К).

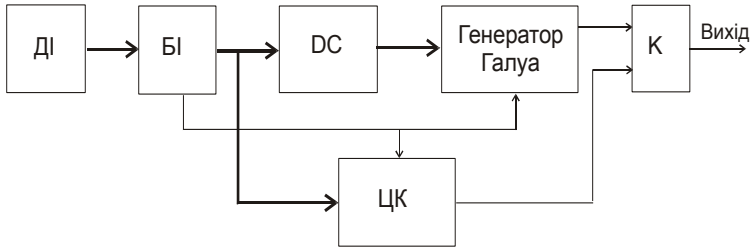


Рис. 15.13. Структурна схема процесора стиснення даних.

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:

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

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. 15.13) складається із регістрів зсуву RG1 і RG2, дешифратора і мультиплексора 16:1. 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.

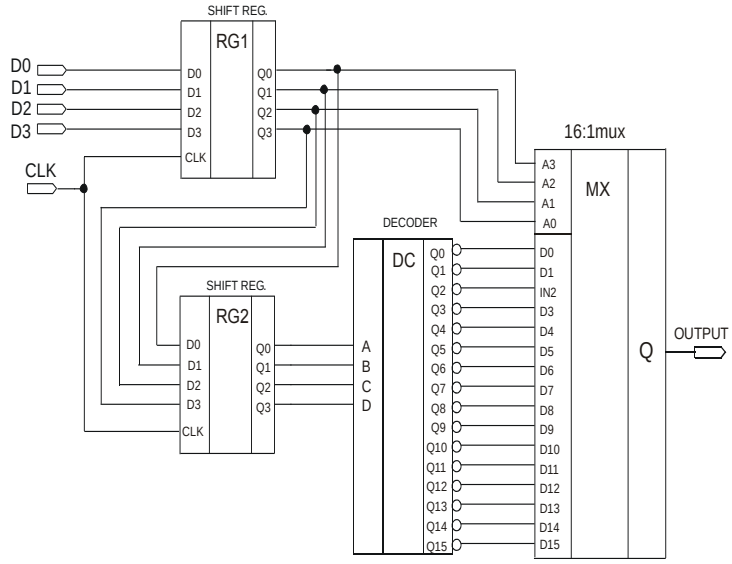


Рис. 15.13. Функціональна схема цифрового компаратора.

Синтезуємо дешифратор, який перетворює двійковий код в код, що відповідає кодону Галуа (табл. 15.3).

За таблицею істинності складаємо рівняння виходів дешифратора:

$$y_1 = \overline{x}_1 \wedge \overline{x}_2 \wedge x_3 \wedge \overline{x}_4 \vee x_1 \wedge \overline{x}_2 \wedge x_3 \wedge x_4 \vee \overline{x}_1 \wedge x_2 \wedge x_3 \wedge \overline{x}_4 \vee$$
$$\vee \overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3 \wedge x_4 \vee x_1 \wedge \overline{x}_2 \wedge \overline{x}_3 \wedge x_4 \vee \overline{x}_1 \wedge \overline{x}_2 \wedge x_3 \wedge x_4 \vee$$
$$\vee \overline{x}_1 \wedge x_2 \wedge x_3 \wedge x_4 ;$$

$$y_2 = x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee$$
$$\vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee$$
$$\vee x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4;$$
$$y_3 = \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee$$
$$\vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee$$
$$\vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4;$$
$$y_4 = x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee$$
$$\vee x_1 \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee$$
$$\vee x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4.$$

Таблиця 15.3.

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:

$$y_1 = \overline{x_2} \wedge x_3 \wedge \overline{x_4} \vee \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_3;$$
$$y_2 = x_1 \wedge x_2 \wedge \overline{x_4} \vee \overline{x_2} \wedge x_3 \wedge x_4 \vee x_1 \wedge x_2 \wedge \overline{x_3} \vee \overline{x_2} \wedge x_4 \wedge$$
$$\wedge (\overline{x_1} \wedge \overline{x_3} \vee x_1 \wedge x_3);$$
$$y_3 = x_2 \wedge \overline{x_4} \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3};$$
$$y_4 = x_1 \wedge \overline{x_3} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_4} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_4}.$$

Згідно отримоної системи логічних рівнянь розроблена функціональна схема дешифратора (рис. 15.14).

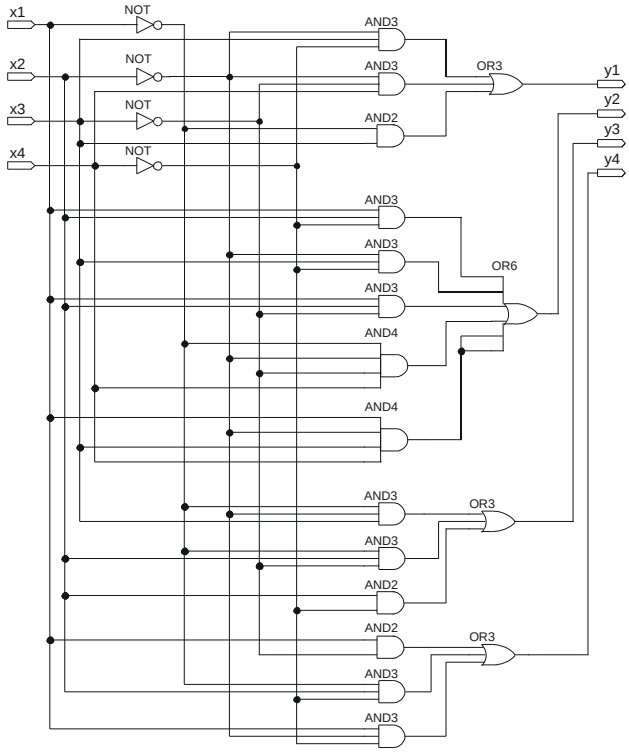


Рис. 15.14. Функціональна схема дешифратора.

Незважаючи на складний математичний апарат, спец процесори базису Галуа характеризуються достатньо простою мікроелектронною реалізацією.

Генератор Галуа реалізований на регістрі зсуву зі зворотнім зв'язком (рис. 15.15). На паралельні інформаційні входи D0–D4 із дешифратора DC поступає чотирирозрядний кодон, який записується синхроімпульсом на вході P/S. З приходом тактових імпульсів на вхід C на виході одержуємо рекурентну послідовність Галуа з періодом $N = 2^4 - 1$.

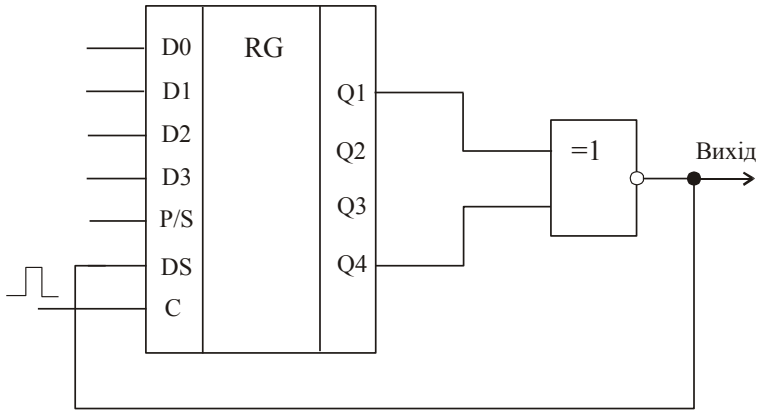


Рис. 15.15. Генератор базисних функцій Галуа.

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.

15.5. 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 станів становить 25 % від їх загальної кількості.

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

Робота спецпроцесора описується наступним алгоритмом (рис.15.16): з формувача ЛСІМ дані у вигляді “0” і “1”, що відповідають стану ДІ і залежать від використаної ЛСІМ, 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:

$$0 \oplus G = G$$
$$1 \oplus G = \overline{G},$$

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. Розрядність непривідних поліномів для генерування рекурентних послідовностей вибираємо 2^r, де $20 \le r \le 30$ 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.

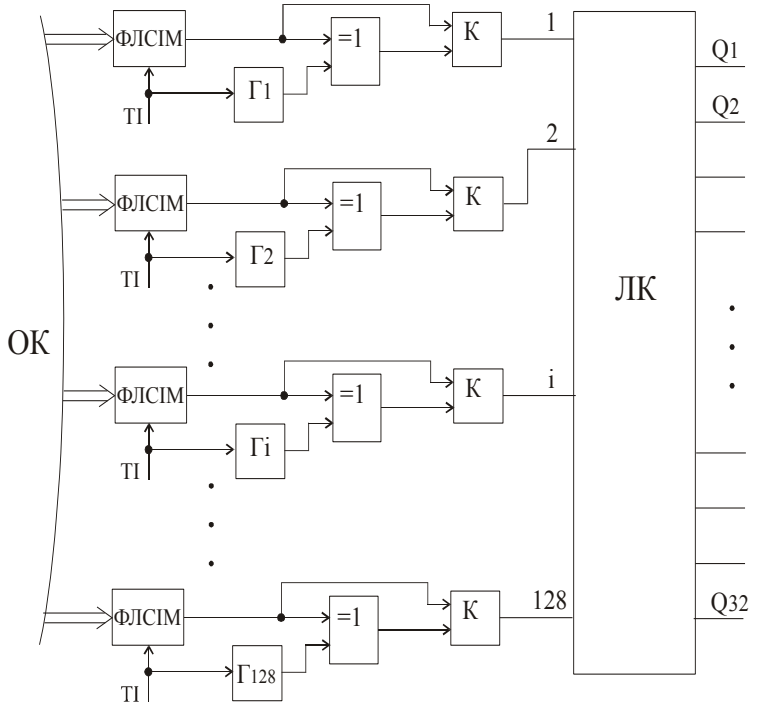


Рис. 15.16. Структурна схема спецпроцесора формування потоків даних ЛСІМ:
ФЛСІМ – формувач даних логіко-статистичних інформаційних моделей; Гі – генератор послідовності Галуа; К – ключ; ЛК – логічний комутатор; ТІ –тактові імпульси.

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

15.6. Спецпроцесори формування сигнальних коректуючих кодів Галуа.

Спецпроцесори даного типу розроблені автором сумісно з Т.М. Гринчишиним.

В загальному випадку функціональна структура спецпроцесора формування потоку даних та організація сигнальних коректуючих кодів описується конвеєрним функціоналом:

$$S_X = F(OI, СП, ОВ),$$

де ОІ – оптичний інтерфейс, який реалізує гальванічну розв'язку комп'ютера або комунікаційного процесора цифрової станції, формує три біт-орієнтовані потоки інформації у вигляді сигналів:

1) синхронізації (SX);
2) "start" :
3) дані стандартного фрейму комп'ютерної мережі "D".

СП – спецпроцесор;

ОВ – оптичний випромінювач.

Функціональні структури модуля ОІ та генератора Галуа показано на рис.15.17, 15.18 відповідно.

Базові структури спецпроцесорів формування сигнальних коректуючих кодів описується наступними функціоналами з конвеєрним виконанням операцій:

1. Функціонал, який характеризує формування ПСК (позиційного сигнального коду), описується виразом:

$$S_X = F(OI, G, M, МП),$$

де G – генератор коду поля Галуа, який формує послідовність 2n-бітів для маніпуляції бітів даних та формування ПСК; М – модулятор, який реалізує формування маніпульованих сигналів, ознак ПСК, МП – мультиплексор.

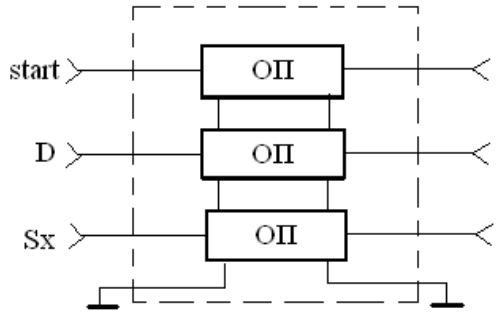


Рис. 15.17. Функціональна структура модуля оптичного інтерфейсу: ОП-оптрон.

Функціональна структура модуля генератора Галуа G показана на рис.15.18.

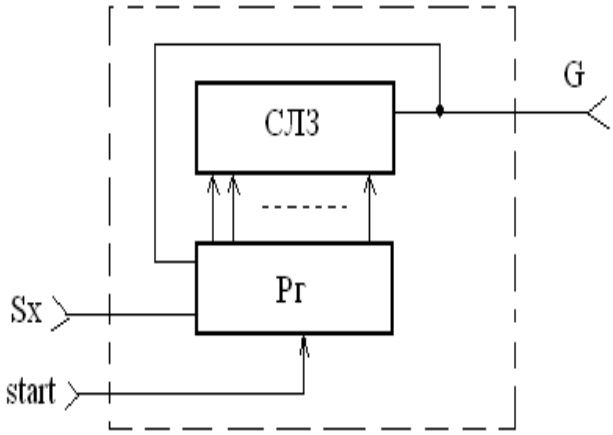


Рис. 15.18. Функціональна структура модуля генератора Галуа:
СЛЗ-схема логічних зв'язків, Рг-регістр.

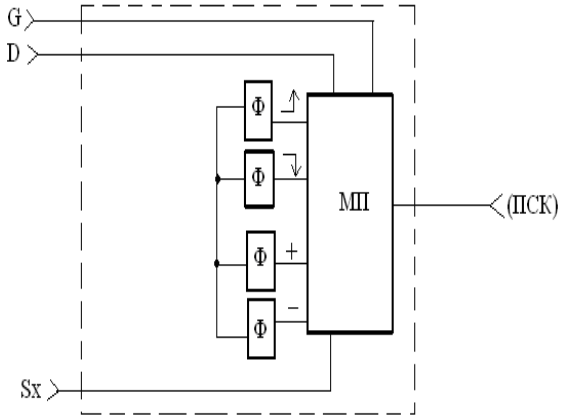


Рис. 15.19. Функціональна структура модулятора ПСК:

1. Ф-формувач відповідного маніпульованого сигналу.
2. Функціонал, згідно якого формується НРСК має вигляд:

$$S_X = F(OI, ЛМ, G_0, МП),$$

де ЛМ – логічний модуль;

G_0 – генератор Галуа сигнальної маніпуляції нульових бітів даних.

Функціональна структура спецпроцесора формування НРСК показана на рис.15.20.

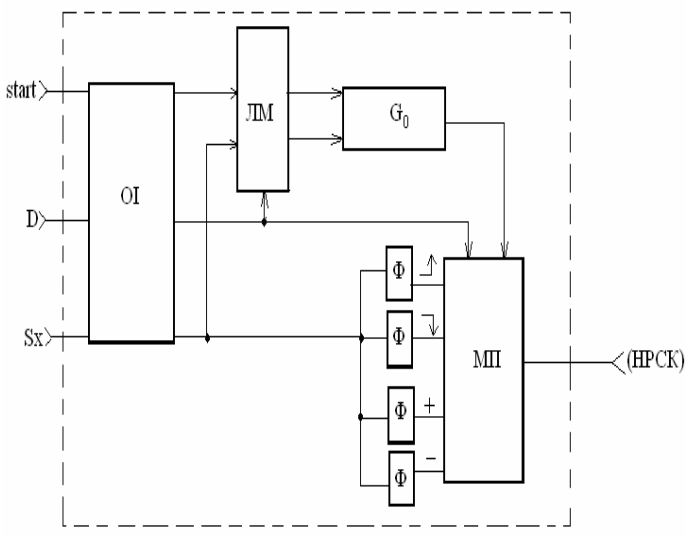


Рис. 15.20. Функціональна структура спецпроцесора формування НРСК.

В даному спецпроцесорі логічний модуль виконує функції синхронізації стартового запуску генератора Галуа, згідно бітів нулів в потоці даних “D”. При цьому одиничні біти потоку даних без додаткового опрацювання через мультиплексор поступають на вихід спецпроцесора, а біти нулів додатково маніпулюються бітами генератора Галуа.

3. Функціонал, згідно якого реалізується формування конвеєрним способом РССК має вигляд:

$$S_X = F(OI, G_0, G_1, ЛМ, МП),$$

де G_0, G_1 – відповідно генератори Галуа маніпульованих даних “0” та “1” .

Функціональна структура спецпроцесора показана на рис.15.21.

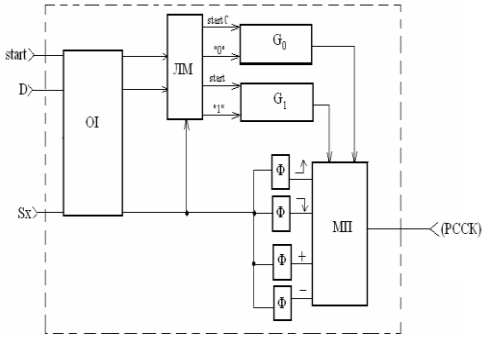


Рис. 15.21. Функціональна структура спецпроцесора формування РССК.

Особливістю структури даного спецпроцесора є симетричне формування Галуа ознак бітів даних. Причому стартування генератора Галуа G_0 та G_1 відбувається згідно появи першого біта нуля або одиниці в потоці даних.

4. Функціонал, згідно якого реалізується формування КССК має вигляд:

$$S_X = F(OI, G_0, G_1, ЛМ, S, ЦА, МП),$$

де S – формувач квазісимвольної ознаки бітової синхронізації ССК;
ЦА – цифровий автомат, що аналізує два поточних біти нульових даних і формує квазісимвольні сигнали “S”.

На рис.15.22. представлено функціональну схему генератора Галуа КССК.

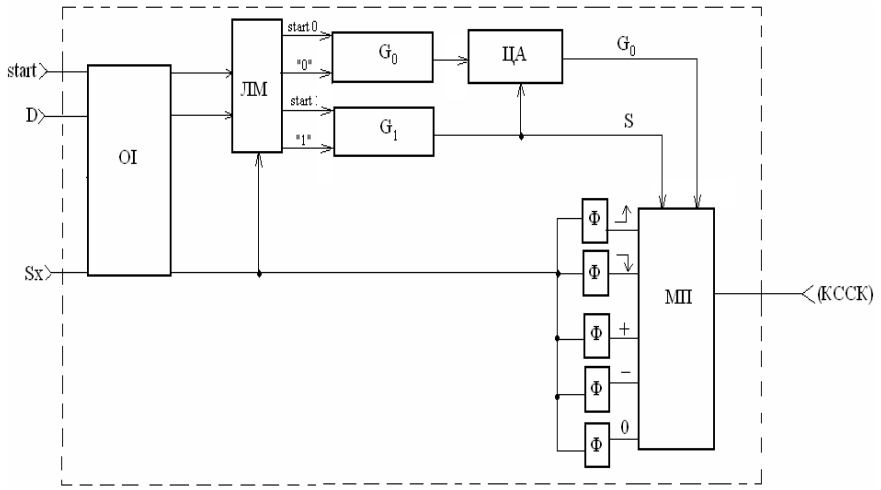


Рис. 15.22. Функціональна структура спецпроцесора формування КССК.

5. Оптичні випромінювачі бісигнального оптичного ІК передавання даних показано на рис.15.23.

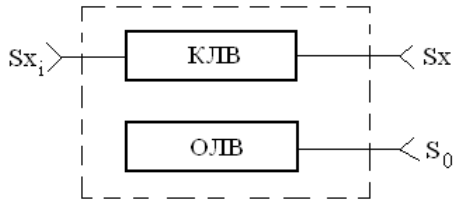


Рис. 15.23. Функціональна структура модуля оптичних випромінювачів:
КЛВ – керований лазерний випромінювач,ОЛВ – опорний лазерний випромінювач.

Аналіз алгоритмічної структурної та часової складності розроблених спецпроцесорів формування сигнальних коректуючих кодів виконується на основі аналітичних виразів оцінки складності, згідно теорії SH-моделей, розробленої М.В. Черкаським. Для розрахунку структурної складності, методом даної оцінки побудована таблиця, яка враховує наявність певного компонента спецпроцесора та експертну оцінку його складності, виходячи з розрахунку об'єму його мікроелектронного обладнання.

Визначення елементарних перетворювачів в програмно-апаратному засобі дозволяє розширити список властивостей і характеристик комп'ютерного алгоритму і включає перетворення деякої сукупності початкових даних у сукупність вихідних даних d: $x_i = \{d_i\} \rightarrow \{d_i\}$.

При цьому елементарний і-ий перетворювач x_i є одиницею апаратної складності і характеризується одиничною часовою складністю: $\forall\ i, l_i = 1$.

При проектуванні мікроелектронних комп'ютерних засобів на основі сучасної елементної бази типу програмовано-логічних матриць ПЛМ, в якості елементарного перетворювача найчастіше вибирають окремий вентиль, який є компонентом логічних елементів та інших більш складних структур процесорних елементів.

Таким чином, згідно визначення існуючого поняття SH-моделі алгоритму апаратна складність визначається сумарною кількістю елементарних перетворювачів і елементів пам'яті деякого рівня апаратних засобів:

$$A=|X|,$$

де X – множина елементів схеми.

Часова складність SH-моделі визначається кількістю елементарних перетворювачів, розташованих вздовж максимального критичного шляху розповсюдження сигналу: $L=|\max X_i|$, тобто максимального сумарного часу затримки сигналів.

Програмна складність визначається логарифмічною мірою ступеня нерегулярності (ентропії) розташування сигналів керування часової діаграми SH-моделі:

$$P = - F \log_2 F/n \cdot m;\ F = \Sigma f_l, \qquad (15.2)$$

де n - кількість входів керування;

m - кількість дискретів часу часової діаграми;

f_l, - кількість сигналів керування l-того фрагмента часової діаграми для обраного рівня ієрархії побудови апаратних засобів;

l - кількість фрагментів часової діаграми, конфігурації яких не повторюються.

Структурна складність алгоритмічного пристрою - це ентропія матриці суміжності:

$$S = - E \log_2 E/ E \cdot n (n-1) , \qquad (15.3)$$

де E - кількість елементів матриці суміжності системи; n - розмір матриці.

Експертна оцінка структурної складності для 15-ти розрядного коду:

ОІ – 1х3 = 3;	ЛМ = 5;
ОВ -1х2 =2;	ЦА = 5 ;
G_0, G_1 = n+3;	МП = 8;
$Ф_1$-$Ф_4$ = 1;	$МП_S$ = 10.
$Ф_S$ = 1;	

Дані занесено в табл.15.4. та показані на діаграмі рис.15.24.

Таблиця 15.4

Таблиця структурної складності спецпроцесорів формування коректуючих кодів.

Тип СП	ОІ	G_0	G_1	ЛМ	$Ф_1$-$Ф_4$	$Ф_S$	ЦА	МП	$МП_S$	ОВ	Σ
ПСК	3	n+3			4			8		2	n+20
НРСК	3	n+3		5	4			8		2	n+25
РССК	3	n+3	n+3	5	4			8		2	n+28
КССК	3	n+3	n+3	5	4	1	5		10	2	n+36

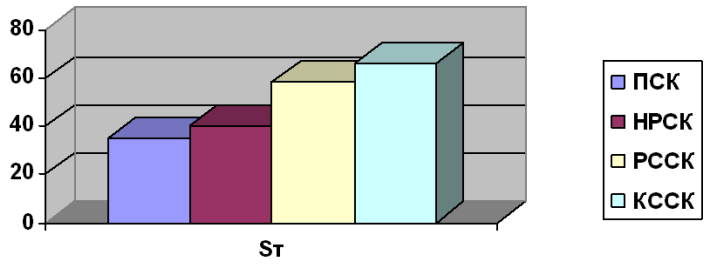


Рис.15.24. Діаграма структурної складності араллельного формування сигнальних араллельн кодів Галуа.

Аналіз діаграми (рис.15.24) показує, що апаратна складність процесорів КССК в порівнянні з ПСК зростає приблизно в два рази. В той же час коректуючі можливості ПСК дозволяють виявляти 75% помилок, а КССК – 100%. Тому згідно критерію ефективності, яка має вигляд:

$$K_e = K_K \cdot V_0 / S_T \cdot C_K , \qquad (15.4)$$

де C_K – вартість експлуатації каналу зв'язку;

V_0 – коефіцієнт зниження швидкодії;

K_K - коефіцієнт корекції помилок;

S_T – структурна складність спецпроцесорів.

K_e (ПСК) = 0.75/35 ·100% = 2.14·0.7 = 1.4;

K_e (КССК) = 1/65 ·100% = 1.53.

Таким чином, згідно експертної оцінки апаратної складності та критерію, який враховує ступінь виявлення та виправлення помилок, а також число повторних передач, ефективність араллельн КССК на 13% вища.

Часова складність розроблених спецпроцесорів розраховується на основі сумарної часової затримки станів при їх конвеєрному опрацюванні компонентами араллельн, згідно виразів:

$\tau_{ПСК} = \tau_{OI} + \tau_{МП} + \tau_{ОВ}$;

$\tau_{НРСК} = \tau_{OI} + \tau_{ЛМ} + \tau_{G} + \tau_{МП} + \tau_{ОВ}$;

$\tau_{РССК} = \tau_{OI} + \tau_{ЛМ} + \tau_{G} + \tau_{МП} + \tau_{ОВ}$;

$\tau_{КССК} = \tau_{OI} + \tau_{ЛМ} + \tau_{G} + \tau_{ЦА} + \tau_{МП} + \tau_{ОВ}$,

де $\tau_{ОВ} = 10V$; $\tau_{МП} = 2V$; $\tau_{ЛМ} = 4V$; $\tau_{G} = 2V$; $\tau_{ЦА} = 4V$; $\tau_{OI} = 10V$,

V – швидкість переключення одного вентиля. Тоді:

$\tau_{ПСК} = 20V+2$;

$\tau_{НРСК} = 20V+8$;

$\tau_{РССК} = 20V+8$;

$\tau_{КССК} = 20V+12$.

Діаграма часової складності процесорів даного класу приведена на рис.15.25.

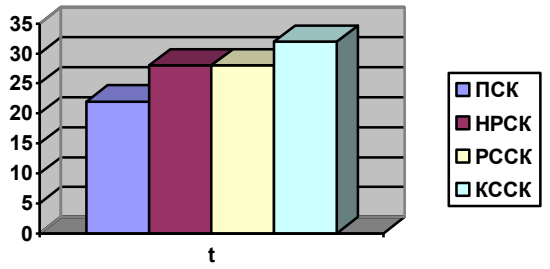


Рис.15.25 Діаграма часової складності спецпроцесорів формування сигнальних коректуючих кодів Галуа.

З діаграми видно, що часова складність різних спецпроцесорів вдрізняється не більше 30% і не залежить від довжини блоку вхідних даних.

15.7. Структура спецпоцесора опрацювання сигналів, рандомізованих в кодах Галуа.

На рис. 15.26 зображено структурну схему спецпроцессора ролзроблено автором сумісно з І.М. Лазаровичем реалізує метод вагового хешування гармонічних сигналів і перетворення їх в широкосмугові рекурентні імпульсні послідовності коду поля Галуа.

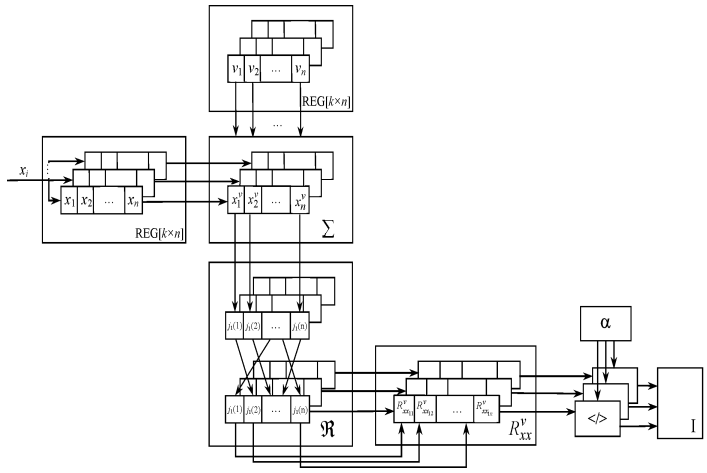


Рис. 15.26. Структурна схема спецпроцесора.

Розглянемо принцип роботу схеми, зображеної на рис. 15.26.

Відліки сигналу x_i поступають на входи m послідовних регістрів зсуву розміром n елементів. Утворюються m вибірок сигналу, які на наступному кроці додаються з ваговими коефіцієнтами, утворюючи масиви значень. На наступному кроці відліки отриманих масивів рандомізуються. Далі обчислюється автокореляційна функція рандомізованого аралле $\boldsymbol{R}_{xx}^{v}$. На останньому кроці роботи схеми виконується логічне порівняння відліків автокореляційної функції з пороговим коефіцієнтом. Результат порівняння демонструється за допомогою індикатора.

З метою спрощення, а отже і підвищення швидкодії в схемі замість автокореляційної функції $\boldsymbol{R}_{xx}^{v}$ доцільно застосувати автоструктурну функцію Колмогорова, яка не містить множення:

$$C_{xx}^{v}(i)=\frac{1}{N}\sum_{k=1}^{n}(x_k-x_{k+i})^2 .$$

Таким чином, алгоритм запропонованого спектрального аналізу реалізується за допомогою операцій додавання, рандомізації, піднесення до квадрату та логічного порівняння і не містить множення, що спрощує його апаратну реалізацію та підвищує швидкодію.

Число арифметичних операцій додавання складає $nm(1+2n)$, операцій перестановок - nm, логічних порівнянь - m, операцій піднесення до квадрату - mn^2.

Перевагою алгоритму спектрального аналізу на основі рандомізації в порівнянні з перетворенням Фур'є, де використовуються операції з комплексними числами, є його простота і висока швидкодія. Окрім цього,

алгоритм працює в реальному часі, тобто при надходженні кожного наступного відліку вхідного сигналу на виході маємо оцінку його спектру. Ще однією з переваг алгоритму є його підвищена завадостійкість за рахунок використання рандомізації.

Основним обмеженням алгоритму є те, що він не дозволяє виявляти спектральні складові в сигналі, який є сумою гармонічних коливань кратних частот.

15.8. Спецпроцесори міжбазисних перетворень Радемахера-Галуа та Галуа-Радемахера.

Даний арал спецпроцесорів реалізується на основі шифраторів та дешифраторів між базисних перетворень.

Синтез 4-х бітового шифратора з десяткової системи числення в базис Галуа виконується згідно табл. 15.5 при зміні вхідного коду від 0 до 9.

Таблиця. 15.5

Відповідність коду Галуа десятковим значенням.

Десяткове значення		код Галуа з захистом від помилок					
		G_5	G_4	G_3	G_2	G_1	G_0
0	x_0	1	1	1	1	0	1
1	x_1	1	1	1	0	1	0
2	x_2	1	1	0	1	0	1
3	x_3	1	0	1	0	1	1
4	x_4	0	1	0	1	1	0
5	x_5	1	0	1	1	0	0
6	x_6	0	1	1	0	0	1
7	x_7	1	0	0	0	1	0
8	x_8	1	0	0	1	0	0
9	x_9	0	0	1	0	0	0

Згідно даної таблиці істиності синтезований шифратор описується системою логічних рівнянь:

$$G_0 = x_0 \vee x_2 \vee x_3 \vee x_6;$$

$$G_1 = x_1 \vee x_3 \vee x_4 \vee x_5;$$

$$G_2 = x_0 \vee x_2 \vee x_4 \vee x_5 \vee x_8;$$

$$G_3 = x_0 \vee x_1 \vee x_3 \vee x_5 \vee x_6 \vee x_9;$$

$$G_4 = x_0 \vee x_1 \vee x_2 \vee x_4 \vee x_6 \vee x_7;$$

$G_5 = x_0 \vee x_1 \vee x_2 \vee x_3 \vee x_5 \vee x_7 \vee x_8$,

де G_0, G_1, …,G_5 біти відповідного кодону Галуа.

На рис. 15.27 показано умовне позначення та принципова схема синтезованого шифратора.

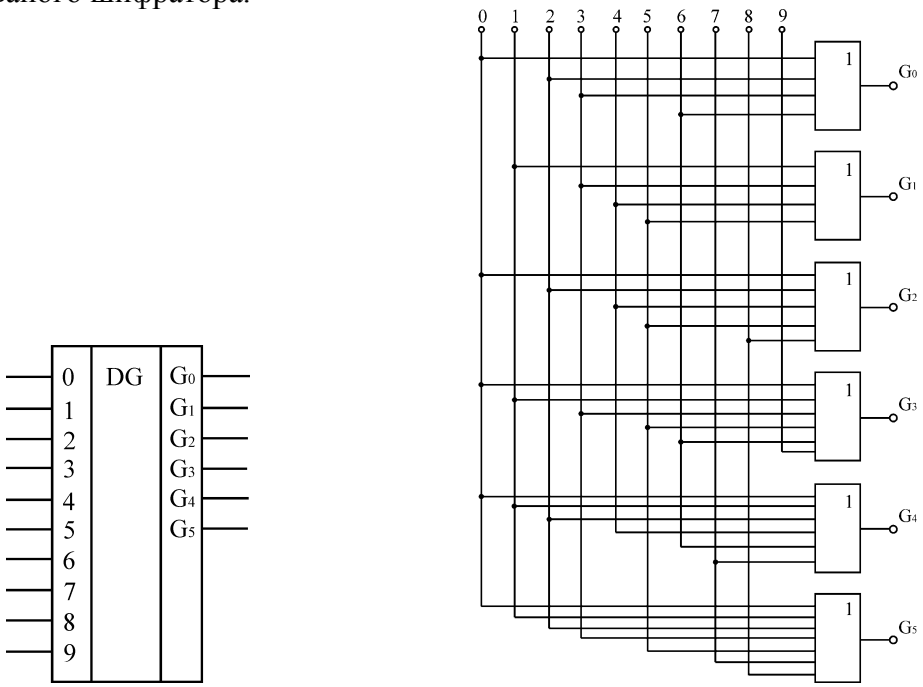


Рис. 15.27. Міжбазисний перетворювач цифр десяткової системи числення в захищений від помилок код поля Галуа. А) – умовне позначення кристала; б) – принципова схема шифратора з десяткової системи числення в базис Галуа.

Позитивною характеристикою аралле шифратора є висока швидкодія, яка визначається часом затримки одного вентиля логічного елементу, а також універсальність, яка обумовлена можливістю зчитування 4-х бітового кодону Галуа без захисту від помилок, або 6-ти бітового кодону з захистом від помилок.

Структура дешифратора з базису Галуа в базис Хаара десяткової системи числення, згідно таблиці істиності (табл. 15.5) описується наступною системою логічних рівнянь:

$$y_0 = G_5 \wedge G_4 \wedge G_3 \wedge G_2; \qquad y_1 = G_5 \wedge G_4 \wedge G_3 \wedge \overline{G}_2;$$

$$y_2 = G_5 \wedge G_4 \wedge \overline{G}_3 \wedge G_2; \qquad y_3 = G_5 \wedge \overline{G}_4 \wedge G_3 \wedge \overline{G}_2;$$

$$y_4 = \overline{G}_5 \wedge G_4 \wedge \overline{G}_3 \wedge G_2; \qquad y_5 = G_5 \wedge \overline{G}_4 \wedge G_3 \wedge G_2;$$

$$y_6 = \overline{G}_5 \wedge G_4 \wedge G_3 \wedge \overline{G}_2; \qquad y_7 = G_5 \wedge G_4 \wedge \overline{G}_3 \wedge \overline{G}_2;$$

$y_8 = G_5 \wedge \overline{G}_4 \wedge \overline{G}_3 \wedge G_2$; $y_9 = \overline{G}_5 \wedge \overline{G}_4 \wedge G_3 \wedge \overline{G}_2$,

і представлена на рис. 15.28.

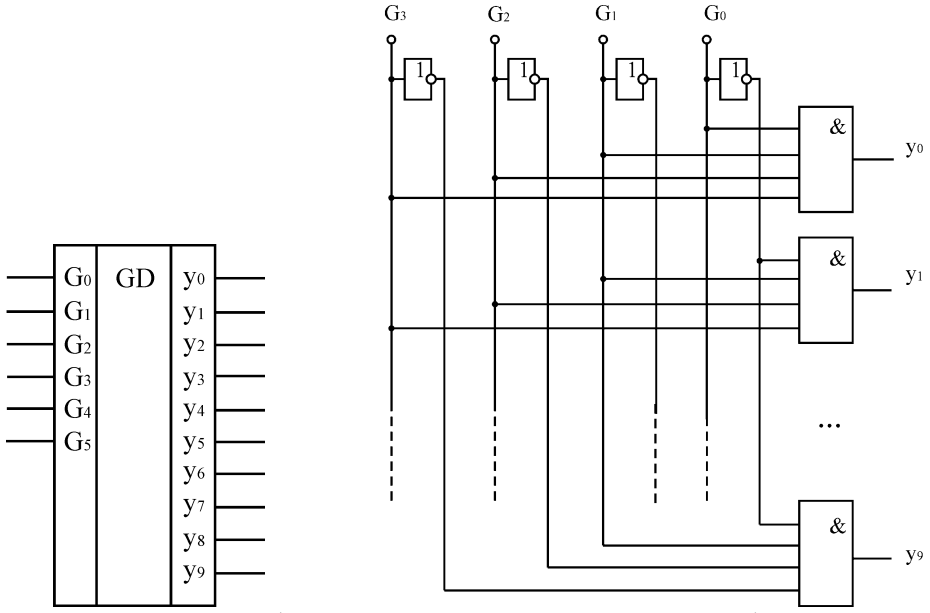


Рис. 15.28. Структура дешифратора з базису Галуа в базис Хаара десяткової системи числення. А) – умовне позначення кристала; б) – структура дешифратора.

Синтез шифратора араллельного коду базису Радемахера в біт орієнтований код базису Галуа виконується згідно таблиці істиності (табл. 15.6) на прикладі 4-х розрядного коду Галуа $GF(\binom{4}{2})$.

Таблиця 15.6.

Таблиця відповідності коду базису Радемахера біт орієнтованому коду базису Галуа

Десяткове значення	Код базису Радемахера				біт орієнтований код базису Галуа
	x3	x2	x1	x0	G
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1

продовження таблиці 15.6

6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Робота даного дешифратора описується наступним логічним виразом:

$G = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2\bar{x}_1x_0 \vee \bar{x}_3\bar{x}_2x_1x_0 \vee \bar{x}_3x_2\bar{x}_1x_0 \vee \bar{x}_3x_2x_1x_0 \vee x_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee x_3\bar{x}_2x_1x_0$,

що після спрощення матиме наступний вигляд: $G = \bar{x}_3x_0 \vee \bar{x}_2\bar{x}_1\bar{x}_0 \vee x_3\bar{x}_2x_1$.

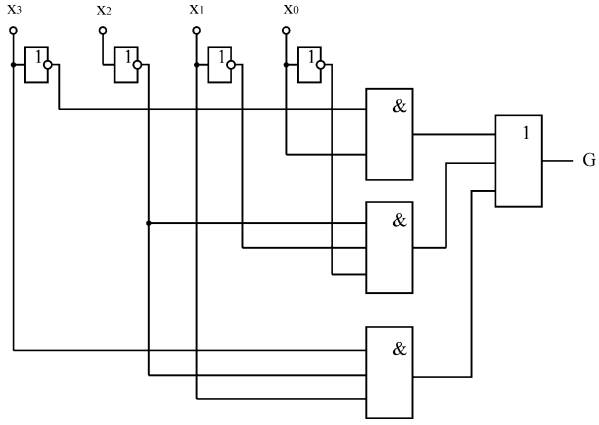


Рис. 15.29. Шифратор паралельного інкрементного коду базису Радемахера в біт орієнтований код базису Галуа

Даний тип шифраторів Галуа, що характеризуються невеликою структурною складністю, яка незначно зростає при збільшенні розрядності k, можна ефективно використати при реалізації перетворювачів паралельних кодів базису Радемахера в біт орієнтовані коди базису Галуа.

Синтез дешифраторів паралельного коду базису Радемахера в базис Галуа класично виконується через проміжне представлення кодів в базисі Хаара.

У загальному випадку при заданому k система логічних рівнянь, яка описує даний дешифратор повинна враховувати відповідні ключі кодів

Галуа. Наприклад для коду Галуа $GF(2^2_4)$ з рекурентним ключем $G_{i+1} = G_i \overline{\oplus} G_{i-3}$ система логічних рівнянь, що описує роботу дешифратора має наступний вигляд:

$$
\begin{aligned}
&G_3 = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2\bar{x}_1x_0 \vee \bar{x}_3\bar{x}_2x_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2x_1x_0 \vee \bar{x}_3x_2\bar{x}_1x_0 \vee \bar{x}_3x_2x_1x_0 \vee \\
&\vee x_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee x_3\bar{x}_2x_1x_0; \\
&G_2 = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2\bar{x}_1x_0 \vee \bar{x}_3\bar{x}_2x_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2x_1\bar{x}_0 \vee \bar{x}_3x_2x_1\bar{x}_0 \vee \bar{x}_3x_2x_1x_0 \vee \\
&\vee x_3\bar{x}_2x_1\bar{x}_0 \vee x_3x_2x_1x_0; \\
&G_1 = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2\bar{x}_1x_0 \vee \bar{x}_3\bar{x}_2x_1x_0 \vee \bar{x}_3x_2x_1\bar{x}_0 \vee x_3\bar{x}_2\bar{x}_1x_0 \vee x_3x_2x_1\bar{x}_0 \vee \\
&\vee x_3x_2x_1x_0; \\
&G_0 = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3\bar{x}_2x_1\bar{x}_0 \vee \bar{x}_3x_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3x_2\bar{x}_1x_0 \vee x_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee x_3x_2\bar{x}_1x_0 \vee \\
&\vee x_3x_2x_1\bar{x}_0 \vee x_3x_2x_1x_0.
\end{aligned}
\tag{15.4}
$$

Мінімізація схеми даного дешифратора приводить до його опису системою логічних рівнянь (15.5) та представлення структурною схемою рис. 15.30.

$$
\begin{aligned}
&G_3 = \bar{x}_1x_3 \vee x_0x_1x_2; \quad G_2 = \bar{x}_2(x_1 \vee \bar{x}_3); \\
&G_1 = x_3(\bar{x}_2\bar{x}_1) \vee x_0; \quad G_0 = \bar{x}_3(\bar{x}_0 \vee x_2).
\end{aligned}
\tag{15.5}
$$

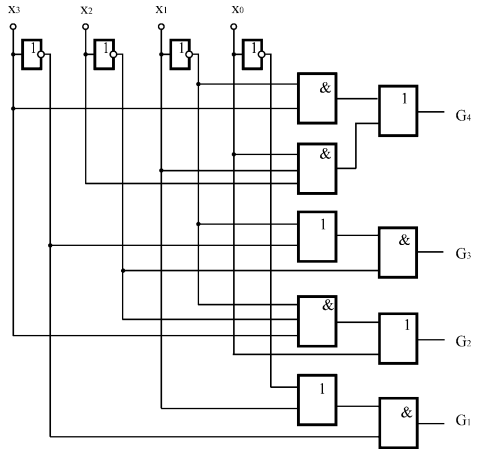


Рис. 15.30. Структура паралельного дешифратора кодів базису Радемахера – Галуа.

Аналіз систем логічних рівнянь типу (15.4),(15.5) для різної розрядності дешифратора *k=4,8,16* показує, що в результаті мінімізації даного типу дешифратора досягається значне зменшення його структурної складності на 40-60% відносно його опису в канонічній формі. Очевидно, що при сучасному рівні мікроелектроніки та застосуванню ПЛМ даний тип

дешифраторів може бути ефективно реалізований для 32 розрядних процесорів з забезпеченням прямого конвертування кодів базису Радемахера в базис Галуа і навпаки, що є особливо актуально при створенні мультибазисних процесорів типу RCG .

Важливим компонентом спецпроцесорів є дешифратор біт орієнтованого коду Галуа в базис Хаара, умовне позначення та структурна схема якого зображена на рис.15.31.

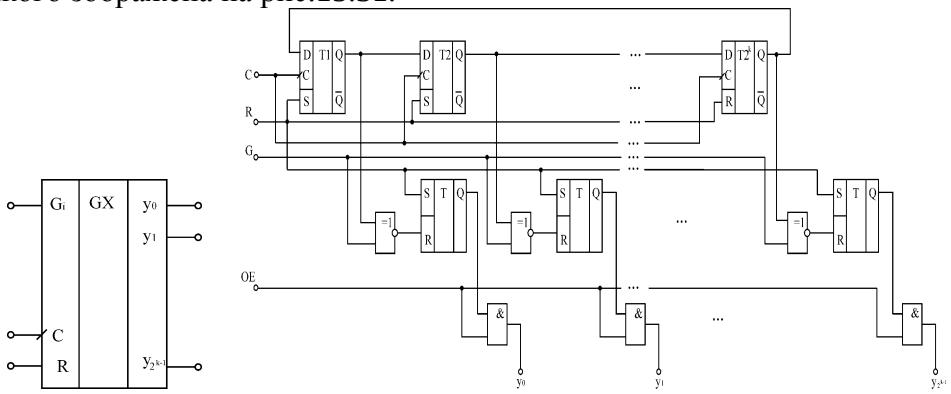


Рис. 15.31. Дешифратора біт орієнтованого коду Галуа в код базису Хаара. а) – умовне позначення кристала; б) – структурна схема.

Робота даного біт орієнтованого дешифратора виконується згідно послідовності мікрокоманд:

1. *R)* $DT_1:=G_1 \wedge DT_2:=G_2 \wedge \ldots \wedge DT_2^k:=G_2^k \wedge ST_i:=1;$

2. C_j) $RT_i:=DT_i \overline{\oplus} G_i ; \quad j \in 1,k ; \; i \in 1,2^k ;$

3. *OE)* $Y_{i-1}:= QT_i; \; i \in 1,2^k .$

Аналіз архітектури даного дешифратора показує, що його структурна складність зростає пропорційно 2^k. В той же час характеризується регулярною архітектурою, яка не залежить від значення k. Час затримки сигналів в дешифраторі даного типу розраховується згідно виразу:

$$T_{дешифратора}=T_т+k(T_т+T_{ле})+T_{ле}.$$

Дешифратори даного класу ефективно використовуються для організації багатопортової ПКД (рис.15.1) на низових рівнях зірково-магістральних РКС.

15.9. Архітектури суматорів у базисі Галуа.

В базисі Радемахера відомі ряд структур суматорів, які показані на рис. 15.32. Робота даних суматорів описується рівняннями:

- для півсуматора $S = A \wedge \overline{B} \vee \overline{A} \wedge B;\ P = A \wedge B;$

- для повного суматора

$$S_i = \overline{A}_i \wedge \overline{B}_i \wedge P_{i-1} \vee \overline{A}_i \wedge B_i \wedge \overline{P}_{i-1} \vee A_i \wedge \overline{B}_i \wedge P_{i-1} \vee A_i \wedge B_i \wedge P_{i-1},$$

$$P_i = \overline{A}_i \wedge B_i \wedge P_{i-1} \vee A_i \wedge \overline{B}_i \wedge P_{i-1} \vee A_i \wedge B_i \wedge \overline{P}_{i-1} \vee A_i \wedge B_i \wedge P_{i-1}.$$

Нарощування розрядності виконується послідовним з'єднанням однорозрядних повних суматорів (рис.15.32. в) в залежності від необхідної розрядності. Основним недоліком суматорів в базисі Радемахера є наявність наскрізного переносу, що приводить до значного зниження швидкодії процесорів, яке пропорційне їх розрядності і часу затримки сигналів згідно виразів:

$$T_{\Sigma R}=kT_c,\ T_c=3T_{ле}. \qquad (15.6)$$

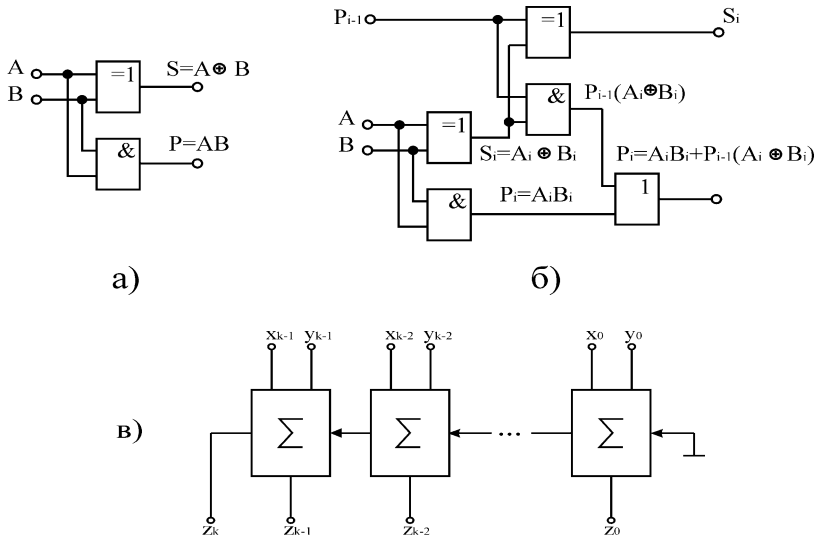


Рис. 15.32. Структури двійкових однорозрядних суматорів.
а) – півсуматор, б) – повний суматор, в) – k розрядний сумматор.

Граничним випадком максимального зниження швидкодії таких суматорів є операція інкременту чи декременту згідно арифметичної операції у базисі Радемахера:

$$\begin{array}{ll} \overleftarrow{111\ldots11} & x_{(2)}=2^{k-1}; \\ +\ 000\ldots01 & y_{(2)}=1\ ; \\ \hline 1000\ldots00 & z_{(2)}=2^{k}\ , \end{array}$$

де ← - символ наскрізного переносу між розрядами суматора.

Слід відмітити, що дана операція інкрементного сумування в базисі Галуа виконується за один такт незалежно від розрядності суматора на базі структури представленої на рис. 15.33, де логічні елементи виконують функції рекурентного формування і+1-х бітів коду Галуа. При цьому час затримки сигналів при виконанні операції інкременту сумування визначається згідно виразу:

$$T_{\Sigma GI}=T_{т}+3kT_{ле}. \qquad (15.7)$$

Перевагою інкрементного суматора в базисі Галуа є наявність пам'яті, в той час як суматор в базисі Радемахера для зберігання даних потребує 3-х k-розрядних регістрів.

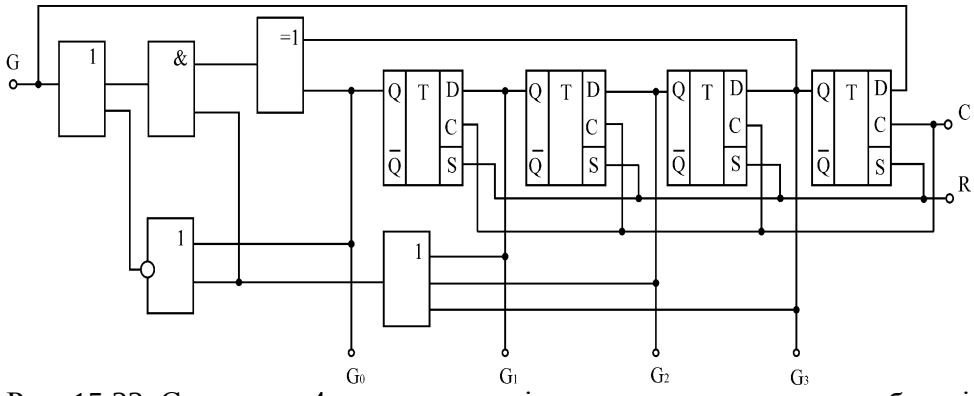


Рис. 15.33. Структура 4-х розрядного інкрементного суматора в базисі Галуа.

Операція сумування кодів у базисі Галуа для суматорів в діапазоні 2^k-*1*, виконується шляхом виконання логічних операцій над бітами коду Галуа першого з доданків згідно таблиці логічних рівнянь, що описують другий доданок. В табл. 15.7 подані формули логічного опису бітів другого доданку, на прикладі 4-х розрядного коду Галуа для суматорів в діапазоні 2^k, що досягається введенням додаткової комбінації "0000" в позиції 12 .

Таблиця 15.7.

Формули логічного представлення кодів Галуа.

Десяткове значення	Код Галуа	Формула суматора			
		b_4	b_3	b_2	b_1
0	1111	b_4	b_3	b_2	b_1

Продовження таблиці 15.7

1	1110	b_3	b_2	b_1	$b_1 \oplus b_4 \oplus b_3$
2	1101	b_3	b_1	b_2	$b_1 \oplus b_4$
3	1010	b_2	b_2	$b_1 \oplus b_4$	$b_1 \oplus b_2 \oplus b_4$
4	0101	b_1	$b_1 \oplus b_4$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$
5	1011	$b_1 \oplus b_4$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$
6	0110	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$	$b_2 \oplus b_3 \oplus b_4$
7	1100	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3$
8	1001	$b_1 \oplus b_2 \oplus b_3$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3$	$b_2 \oplus b_4$
9	0010	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3$	$b_2 \oplus b_4$	$b_1 \oplus b_3 \oplus b_4$
10	0100	$b_1 \oplus b_3$	$b_2 \oplus b_4$	$b_1 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2$
11	1000	$b_2 \oplus b_4$	$b_1 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2$	$b_2 \oplus b_3$
12	0000	$b_1 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2$	$b_2 \oplus b_3$	$b_3 \oplus b_4$
13	0001	$b_1 \oplus b_2$	$b_2 \oplus b_3$	$b_3 \oplus b_4$	b_4
14	0011	$b_2 \oplus b_3$	$b_3 \oplus b_4$	b_4	b_3
15	0111	$b_3 \oplus b_4$	b_4	b_3	b_2

Такий спосіб опису функцій суматора в базисі Галуа передбачає емуляцію його роботи програмним шляхом, а також виконання операції сумування на основі матриці коефіцієнтів d_{ij} (табл. 15.8), яка використовується для логічного формування бітів коду Галуа суми доданків згідно виразу:

$$b_i = d_{i,k} \cdot b_k \oplus d_{i,k-1} \cdot b_{k-1} \oplus ... \oplus d_{i,1} \cdot b_1 .$$

Таблиця 15.8.

Матриця коефіцієнтів d_{ij}.

Десяткове значення	Код Галуа	Формула суматора			
		d_{j4}	d_{j3}	d_{j2}	d_{j1}
0	1111	1000	0100	0010	0001
1	1110	0100	0010	0001	1001
2	1101	0010	0001	1001	1011
3	1010	0001	1001	1011	1111
4	0101	1001	1011	1111	0111
5	1011	1011	1111	0111	1110
6	0110	1111	0111	1110	0101

Продовження таблиці 15.8

7	1100	0111	1110	0101	1010
8	1001	1110	0101	1010	1101
9	0010	0101	1010	1101	0011
10	0100	1010	1101	0011	0110
11	1000	1101	0011	0110	1100
12	0000	0011	0110	1100	1000
13	0001	0110	1100	1000	0100
14	0011	1100	1000	0100	0010
15	0111	0111	1110	1101	0011

Розглянемо приклад виконання операції додавання двох чисел в базисі Галуа на основі матриці коефіцієнтів d_{ij}. Нехай $X_{(10)}=2;\ Y_{(10)}=5$, тоді $X_G=1101;\ Y_G=1011$. Тобто X_G відповідає коду $b_4=1;\ b_3=1;\ b_2=0;\ b_1=1$, а код Y_G згідно табл. 15.8 відповідає логічним операціям над бітами X_G: $b_1 \oplus b_2 \oplus b_4$; $b_1 \oplus b_2 \oplus b_3 \oplus b_4$; $b_1 \oplus b_2 \oplus b_3$; $b_2 \oplus b_3 \oplus b_4$, що відповідає кодам d_{ij} 1011; 1111; 0111; 1110 з табл. 2.14. Тобто, результат сумування даних чисел виконується за допомогою логічної обробки кодів X_G та коефіцієнтів d_{ij}, які відповідають коду Y_G:

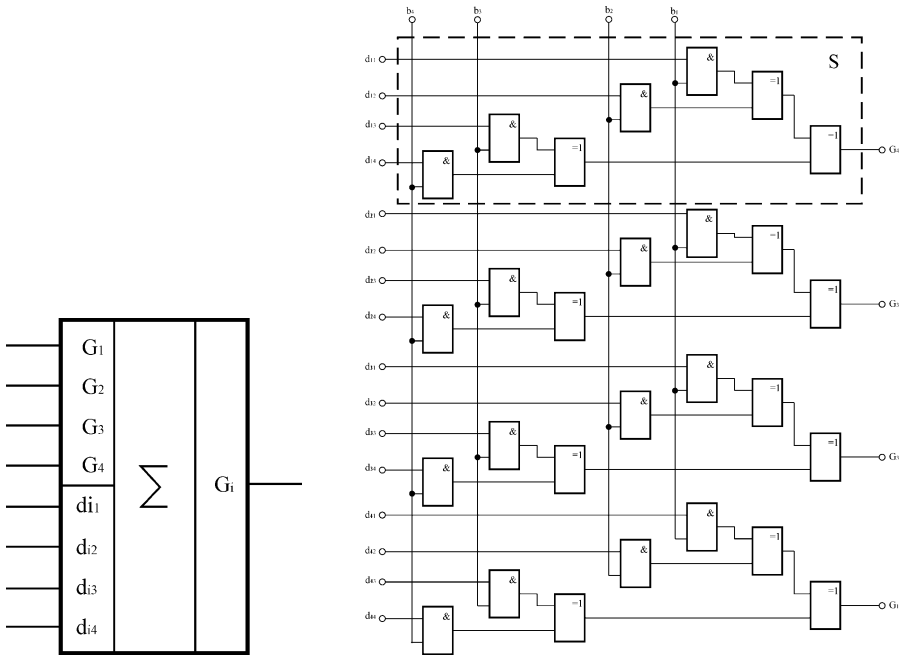
$$G_4=1\wedge b_4 \oplus 0\wedge b_3 \oplus 1\wedge b_2 \oplus 1\wedge b_1 = 1\wedge 1\oplus 0\wedge 1\oplus 1\wedge 0\oplus 1\wedge 1=1;$$

$$G_3=1\wedge b_4 \oplus 1\wedge b_3 \oplus 1\wedge b_2 \oplus 1\wedge b_1 = 1\wedge 1\oplus 1\wedge 1\oplus 1\wedge 0\oplus 1\wedge 1=1;$$

$$G_2=0\wedge b_4 \oplus 1\wedge b_3 \oplus 1\wedge b_2 \oplus 1\wedge b_1 = 0\wedge 1\oplus 1\wedge 1\oplus 1\wedge 0\oplus 1\wedge 1=0;$$

$$G_1=1\wedge b_4 \oplus 1\wedge b_3 \oplus 1\wedge b_2 \oplus 0\wedge b_1 = 1\wedge 1\oplus 1\wedge 1\oplus 1\wedge 0\oplus 0\wedge 1=0.$$

Отримана система логічних рівнянь дозволяє синтезувати структуру 4-х бітового суматора Галуа, який зображений на рис. 15.34.



а) б)

Рис. 15.34. Структура суматора кодів Галуа.
а) – умовне позначення кристала однорозрядного суматора; б) – структурна схема 4-х розрядного сумматора.

Аналіз структури операційного пристрою сумування в базисі Галуа показує, що він характеризується регулярною архітектурою з часом затримки сигналів:

$$T_{\Sigma G}=3T_{ле}+(k/4)T_{ле}. \quad (15.8)$$

Для функціонування даного операційного пристрою необхідно виконувати дешифрування коду k-розрядного коду другого доданку в k^2-розрядний код коефіцієнтів d_{ij}. На рис. 15.35 – зображено умовне позначення (а) та розроблена канонічна структура дешифратора суматора Галуа (б), яка характеризується часом затримка згідно виразу:

$$T_{д\Sigma}=3T_{ле}. \quad (15.9)$$

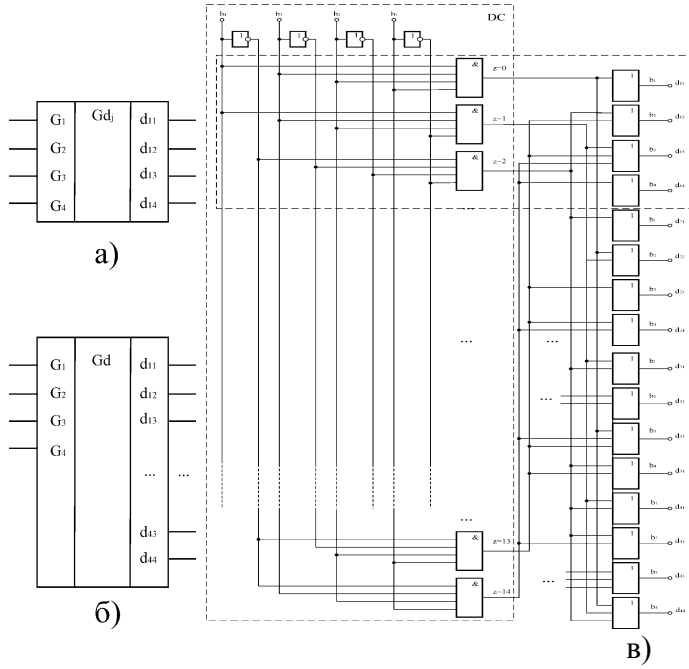


Рис.15.35. Дешифратор Галуа а), б) – умовні позначення кристалів однорозрядного та 4-х розрядного сумматорів; в) – структура дешифратора 4-х розрядного суматора.

При використанні умовних позначень компонентів суматора Галуа (рис. 15.34 а, 15.35 б), отримаємо структурну схему 4-х розрядного паралельного суматора Галуа, яка показана на рис. 15.36.

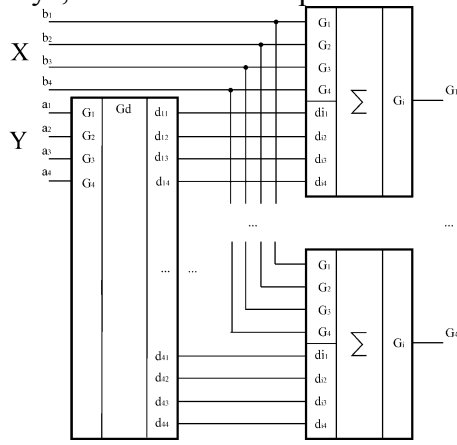


Рис. 15.36. Структурна схема 4-х розрядного паралельного суматора Галуа.

При використанні k-х розрядних фрагментів дешифраторів (рис. 15.35 а) та одно розрядних суматорів (рис. 15.34 а) структура паралельного суматора k-розрядного паралельного суматора Галуа отримає вид (рис. 15.37).

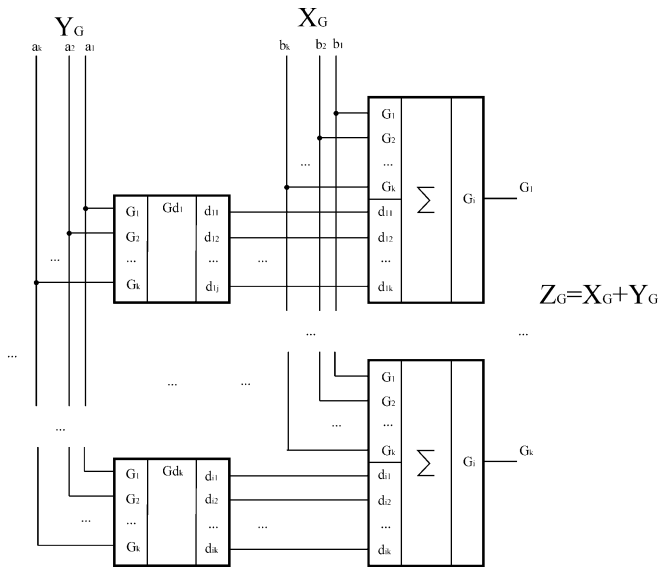


Рис. 15.37. Структурна схема k розрядного паралельного суматора Галуа

Згідно виразів (15.8), (15.9) час затримки сигналів при виконання операції сумування в базисі Галуа на основі структури рис.15.37 дорівнює:

$$T_{\Sigma G}=5T_{ле}+kQ_{ле}, \qquad (15.10)$$

де $Q_{ле}=T_{ле}/4$, оскільки дана затримка відповідає 4-м розрядам суматора, а при подвоєні розрядності суматора в схему послідовно включається один додатковий елемент "виключаюче АБО"; затримка сигналів $5T_{ле}$, яка включає 3 послідовні елементи однорозрядного суматора Галуа та 2 елементи дешифратора.

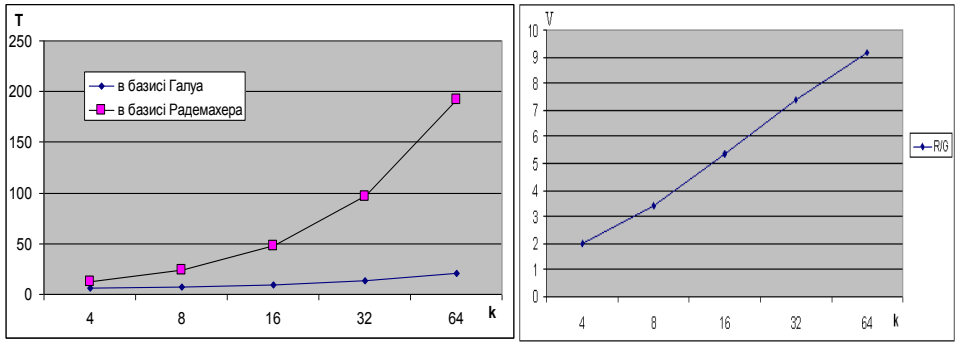
Оцінка швидкодії розраховується за формулами:

$$V_{\Sigma R}=\frac{1}{3kT_{ле}},\ V_{\Sigma G}=\frac{1}{5T_{ле}+kQ_{ле}}.$$

Звідки відносна оцінка швидкодії суматора в базисі Галуа в порівнянні з реалізацією суматора в базисі Радемахера (15.6) розраховується згідно виразу:

$$V = \frac{3kT_{ле}}{5T_{ле} + kQ_{ле}}.$$

Характеристики швидкодії суматорів у базисі Радемахера-Галуа представлені на рис.15.38.



а) б)

Рис. 15.38. Характеристики швидкодії суматорів у базисах Радемахера та Галуа в залежності від розрядності процессора. а) – характеристики часу затримки паралельних суматорів; б) – відносна оцінка швидкодії.

Оскільки діапазон кодування чисел в суматорах Галуа не перевищує 2^k обов'язковою вимогою правильної роботи даного класу суматорів без переповнення розрядної сітки є умова $X_G+Y_G<2^k$.

З графіка (рис.15.38. б) видно, що при побудові процесорів з розрядністю k=16-64 швидкодія суматорів в базисі Галуа перевищує швидкодію суматорів в базисі Радемахера в 5-9 разів.

15.10. Структурна схема багатоканального спецпроцесора базису Галуа з вихідними шумоподібними сигналами.

Структурна схема процесора даного класу реалізована конвеєрної архітектери з багатоканальним АЦП Галуа приведена на рис. 15.39.

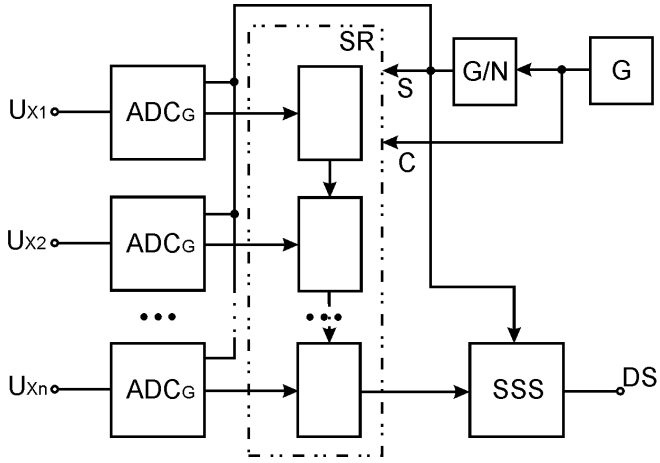


Рис. 15.39. Структура багатоканального спецпроцесора в базисі Галуа з вихідними шумоподібними сигналами.

На рис. 15.39 символами ADC_G; SR; G/N; G; SSS позначені відповідно елементи АЦП Галуа; регістр зсуву; подільник частоти; генератор тактових імпульсів та генератор дискретних широкосмугових кодових послідовностей (ДШКП).

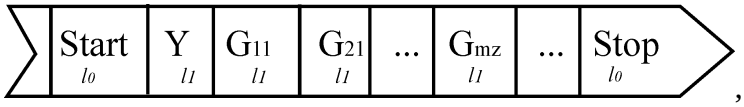
Частота слідування інформаційних даних задається генератором G, подільник частоти G/N (де N кількість каналів АЦП) задає частоту формування вихідних сигналів з АЦП. Формувач вихідних ШКП SSS замінює кожен інформаційний біт відповідною двовимірною ШКП та вставляє на початку і в кінці кожного нового перетворення спеціальний старт-стопний ШКП кодову послідовність, яка дозволяє розпізнати початок та кінець інформаційних даних. Таким чином в спецпроцесорі використовується три ШКП, які ідентифікують адрес та дані АС оснащеного спецпроцесором. Один ДШКП використовується в якості старт-стопних сигналів, а два інші для кодування нулів та одиниць біт-орієнтованого потоку даних. Довжина пакету даних, що формується на виході спецпроцесора визначається числом каналів та розрядністю кодів в кожному каналі відповідно матриці даних

$$\begin{vmatrix} G_{1,1} & G_{2,1} & \dots & G_{i1} & \dots & G_{m1} \\ G_{2,1} & G_{2,2} & \dots & G_{i1} & \dots & G_{m2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ G_{1j} & G_{2j} & \dots & G_{ij} & \dots & G_{mj} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ G_{1z} & G_{2z} & \dots & G_{iz} & \dots & G_{mz} \end{vmatrix}, \qquad (15.11)$$

де в кожній стрічці відповідно представлені і-ті біти Галуа (1, 2,…, z), які паралельно скануються на виході багатокального АЦП Галуа.

При цьому розрядність кодів, які представлені відповідними стовпцями матриці (15.11) $z=k+n$, де k – розрядність ся БАП, n, число захисних бітів коду Галуа.

В результаті на виході АЦК формується біт-орієнтований потік даних, який містить m×z бітів Галуа. Даний спосіб формування вихідних даних забезпечує рандомізацію та певний рівень захисту від пачок помилок, що виникають під дією концентрованих завад в трактах міжпроцесорного обміну даними. З врахуванням старт-стопних сигналів та кодування кожного біту двовимірним ШКП на виході спецпроцесора формується фрейм даних у вигляді:



,

де Y – байт коду, який ідентифікує тип фрейма, l_0, l_1 – відповідно розрядність ДШКП, що кодує старт-стопні сигнали та інформаційні біти фрейма. Отже на виході спецпроцесора періодично або ініціативно формуються пакети даних довжиною $N = 2L_0 + 8l_1 + m \cdot z \cdot l_1$ чіпів.

Наприклад: для параметрів АЦК $m=8$; $k=10$; $n=2$; l_0=50; l_1=32 об'єм пакету даних, який формується на виході СП буде рівним 3428 чіпів. В результаті двовимірної кореляційної обробки даної послідовності чіпів на виході приймального процесора формуються сигнали показані на рис. 15.40.

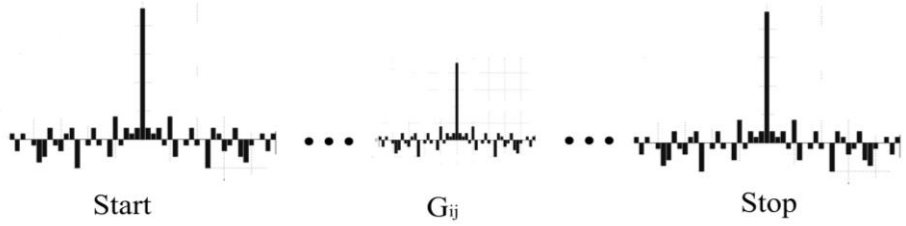


Рис. 15.40. Сигнали на виході кореляційного приймача даних фрейма, який формується СП.

При стандартній ширині смуги частот, яка виділяється для DSSS каналів 22МГц, максимальна тактова частота БАЦП АС буде складати 3,2кГц, що дозволяє створити безпровідний дистанційний передавач 4-х каналів цифрового телефонного зв'язку, або 8 каналів 10-бітних каналів технологічних даних.

При використанні стандарту 108 Мбіт мережі КС і застосуванні 16 бітного двовимірного коду при однаковій завадозахищеності може бути досягнутий стабільний трафік передавання даних на швидкості не менше 150 Мбіт.

В системах з низькою швидкодією передавання даних в границях 1-10 біт/с, які характерні для систем охоронної сигналізації, екологічного моніторингу навколишнього середовища, можуть бути ефективно використані СП даного класу в складі АС, шляхом використання ДШКП великої розрядності ($l \geq 16 \times 16$).

Використання різних ансамблів ШКП для кодування вихідних даних СП дозволяє реалізувати кодове розділення каналів зв'язку та асинхронний режим функціонування автономних сенсорів на низових рівнях РКС.

15.11. Структурна схема спецпроцесора на основі інтегрально-імпульсної технології у базисі Галуа.

Структура реалізованого одноканального спецпроцесора в базисі Галуа базується на конвеєрній топології представленій на рис. 15.39 і зображена на рис. 15.41.

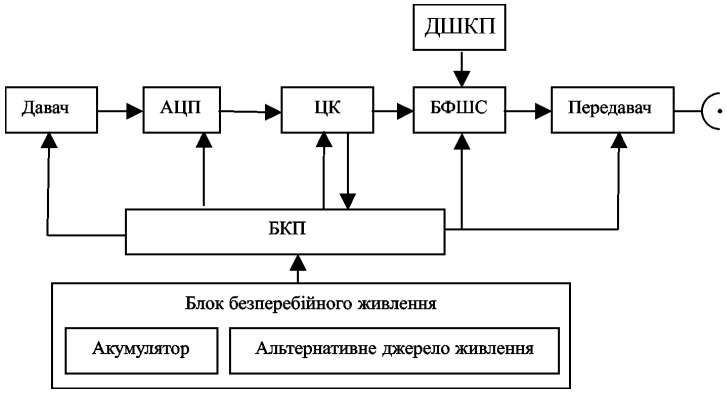


Рис. 15.41. Структура одноканального спецпроцесора в базисі Галуа.

Даний пристрій складається з таких основних структурних вузлів:

- давач – джерело інформації (мікрофон, детектор руху, давач тиску, термопара, перемикач та інші);

- АЦП – типу перетворювач напруга-частота, який здійснює перетворення аналогових даних в потік частотно-модульованих імпульсів (кристал фірми Analog Device типу AD654; живлення 5В; діапазон вихідних частот 500кГц; споживана потужність 2 мА);

- ЦК – цифровий кодер, здійснює перетворення послідовності бітів в послідовність бітів Галуа, а також цифрового інтегрування вхідної інформації, реалізований на основі послідовного з'єднаного подільника частоти на базі 20-бітного лічильника Галуа з зовнішньо комутованим ключем та коефіцієнтом ділення 3,7,15 і т.д., та 20 розрядного лічильника Галуа з періодом 2^{20-1} і кодовим ключем $x_1 \oplus x_{18}$.

- БФШС – блок формування шумоподібного сигналу, здійснює заміну кожного біта відповідним ДШКП розмірністю 3×3, 3×5, 4×4, 4×5, 5×5);

- БШПК – блок унікальних в системі трьох еталонних двовимірних ШКП, які використовуються для заміни старт-стопних інформаційних бітів Галуа *l*-розрядними послідовностями чіпів;

- передавач – широкосмуговий радіопередавач (на базі кристалу фірми Freescale типу MC21213);

- БКП – блок керування подіями здійснює синхронізацію роботи компонентів СП;

- блок безперебійного живлення – здійснює безперебійне енергопостачання АС, містить в собі акумуляторну батарею та альтернативне джерело живлення (сонячну батарею, термобатарею і т.п.).

Компоненти одноканального СП ЦК та БШПК реалізуються на базі ПЛМ (фірми Altera серії MAX7000).

15.12. Реалізація компонентів процесорів Галуа на ПЛМ.

15.12.1. Паралельний суматор в базисі Галуа.

Для апаратної реалізації паралельного суматора Галуа запропонованого у другому розділі формуємо базовий примітив (рис. 15.42).

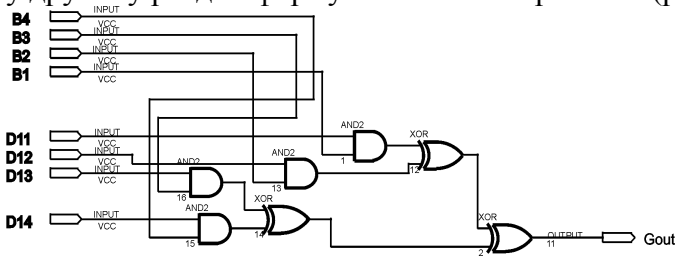


Рис. 15.42. Базовий примітив суматора Галуа: B1-B4 – входи операндів в коді Галуа; D11-D14 – входи кодів з матриці коефіцієнтів; Gout – вихід операції сумування в базисі Галуа.

Даний базовий примітив паралельного суматора є однорозрядним 4-ох бітовим суматором Галуа. Для отримання повного чотирьох-розрядного суматора проводимо збільшення кількості каскадів (рис.15.43).

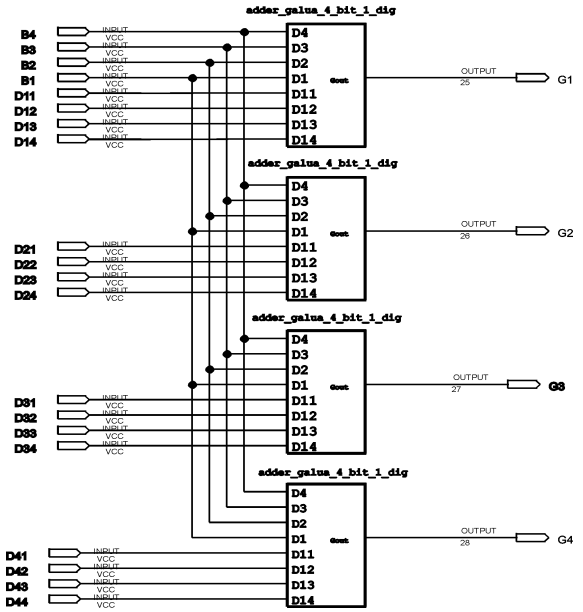


Рис. 15.43. Реалізація 4-ох розрядного суматора Галуа: B1-B4 – входи операндів в коді Галуа; D11-D44 – входи кодів з матриці коефіцієнтів; G1-G4 – виходи бітів операції сумування в базисі Галуа**.**

Проведений порівняльний аналіз реалізації паралельного суматора Галуа представлено у табл. 15.9.

Таблиця. 15.9.

Дослідження системних характеристик паралельних суматорів Радемахера та Галуа.

			Розрядність						
Суматор Галуа	Макро-ком	I/O	8		16		32		Макс. част. МГц
			Макро-ком.	I\O	Макро-ком.	I\O	Макро-ком.	I\O	
			40	76	80	152	160	304	
EPM3032ALC44-4	32	30	125,0	253,3	250	506,7	500	1013,3	157,13
EPM7032LC44-6	32	32	125,0	237,5	250	475	500	950	193,17
EPM9320LC84-15	320	56	12,5	135,7	25	271,4	50	542,9	86,43
EPF6010ATC100-1	880	67	4,5	113,4	9,1	226,9	18,2	453,7	69,93
EPF8282ALC84-2	208	64	19,2	118,8	38,5	237,5	76,9	475	64,52
EPF10K30ETC144-1	1728	96	2,3	79,2	4,6	158,3	9,3	316,7	81,97

Продовження таблиці 15.9

Суматор Радемахера	Макро-ком	I/O	Розрядність						Макс. част. МГц
			8		16		32		
			Макро-ком.	I\O	Макро-ком.	I\O	Макро-ком.	I\O	
			10	26	27	50	64	98	
EPM3032ALC44-4	32	30	31,25	86,66667	84,375	166,67	200	326,667	30,58
EPM7032LC44-6	32	32	31,25	81,25	84,375	156,25	200	306,25	41,67
EPM9320LC84-15	320	56	3,125	46,42857	8,4375	89,285	20	175	13,89
EPF6010ATC100-1	880	67	1,136	38,80597	3,0681	74,626	7,2727	146,269	12,99
EPF8282ALC84-2	208	64	4,807	40,625	12,980	78,125	30,769	153,125	10,31
EPF10K30ETC144-1	1728	96	0,578	27,08333	1,5625	52,083	3,7037	102,083	22,73

На основі даних табл. 15.9 проведемо оцінку частоти роботи паралельних суматорів Радемахера та Галуа реалізованих на ПЛІС (рис.15.44).

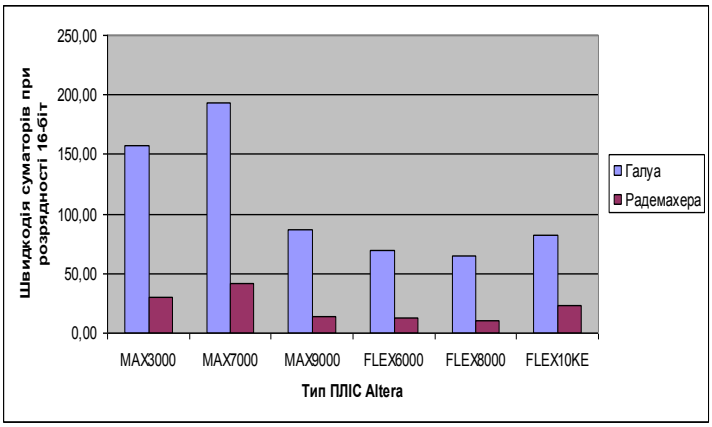


Рис. 15.44. Графік максимальної частоти роботи паралельних суматорів Радемахера та Галуа.

З рис. 15.44 видно, що паралельні суматори Галуа мають значно вищу швидкодію ніж класичні паралельні суматори однакової розрядності базису Радемахера. При чому для кристалів ПЛІС невеликої ємності ця перевага більша по відношенню до кристалів великої ємності. Максимальне відношення швидкостей роботи суматорів в базисі Галуа до суматорів в базисі Радемахера спостерігається при їх реалізації на кристалах ПЛІС MAX7000 та MAX9000 відповідно в 4,6 та 6,2 рази.

Нижчі показники швидкодії суматорів реалізованих по відношенню до теоретичних розрахунків обумовлені універсальністю комірок ПЛІС та особливістю САПР Max+plus II та VHDL.

15.12.2. Лічильники - формувачі ШКП.

При реалізації лічильників Галуа на ПЛІС проведено дослідження їх структури, яке показало, що лічильники даного типу мають регулярну структуру і складаються з однотипних структурних компонентів. На основі даного компоненту спроектовано базовий примітив (рис. 15.45).

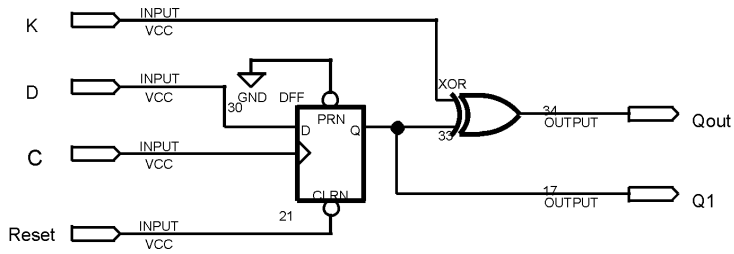


Рис. 15.45. Базовий примітив лічильника Галуа: Reset – скидання лічильника в 0; D – інформаційний вхід на тригер регістра зсуву; C – вхід синхронізації; K – вхід ключової послідовності; Qout – вихід з "виключаюче АБО"; Q1 – прямий вихід і-го тригера

Використовуючи базовий примітив, який являє собою повноцінний однорозрядний лічильник Галуа, здійснюємо нарощення числа розрядів шляхом збільшення кількості елементів (рис. 15.46).

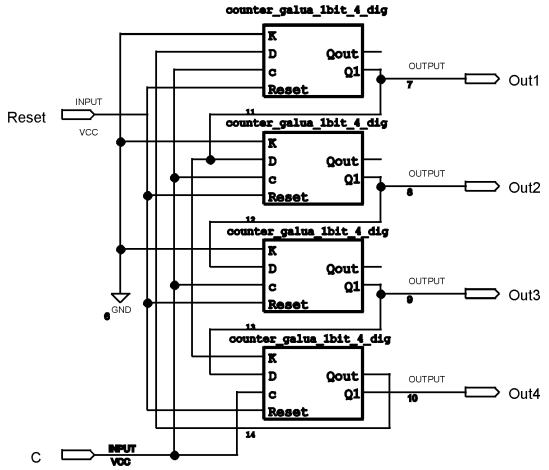


Рис 15.46. Реалізація 4-х розрядного лічильника Галуа: Reset – скидання лічильника в 0; D – інформаційний вхід на тригер регістра зсуву; C – вхід синхронізації; K – вхід ключової послідовності; Qout – вихід елемента "виключаюче АБО"; Q1 – вихід і-го тригера; Out1-Out4 – виходи бітів паралельного коду Галуа

Як видно з рис. 15.46, при збільшенні кількості розрядності лічильника Галуа зростає лише кількість виходів бітів коду лічильника (Out1-Out4) , а число службових входів залишається сталим.

При реалізації лічильників Галуа розрядностей 8-32 біт на ПЛІС фірми Altera проведено дослідження кількості макрокомірок та виводів для різних сімейств ПЛІС.

В результаті досліджень встановлено, що для сімейств MAX3000, MAX7000, MAX9000, FLEX6000, FLEX8000, FLEX10K, один примітив лічильника Галуа використовує одну макрокомірку та 6 виводів. При збільшенні розрядності кількість макрокомірок і виводів збільшується лінійно. Результати досліджень системних характеристик синхронних лічильників представлено у табл.15.10.

Таблиця 15.10.

Дослідження системних характеристик синхронних лічильників Джонсона, Радемахера, Галуа.

			Розрядність						
Джонсон	Число макро-комірок	Число входів	8		16		32		Частота МГц
			Макро-ком. %	I\O %	макроком. %	I\O %	Макро-ком. %	I\O %	
			8	10	16	18	32	34	
EPM3032ALC44-4	32	30	25	33,3	50	60	100	113,3	227,2
EPM7032LC44-6	32	32	25	31,2	50	56,2	100	106,2	303,3
EPM9320LC84-15	320	56	2,5	17,8	5	32,1	10	60,7	117,6
EPF6010ATC100-1	880	67	0,9	14,9	1,8	26,8	3,6	50,7	111,1
EPF8282ALC84-2	208	64	3,8	15,6	7,6	28,1	15,3	53,1	96,1
EPF10K30ETC144-1	1728	96	0,4	10,4	0,9	18,7	1,8	35,4	250

			Розрядність						
Повний Галуа	Число макро-комірок	Число входів	8		16		32		Частота МГц
			Макроком %.	I\O %	Макро-ком. %	I\O %	Макро-ком. %	I\O %	
			9	10	17	18	33	34	
EPM3032ALC44-4	32	30	28,1	33,3	53,1	60	103,1	113,3	227,2
EPM7032LC44-6	32	32	28,1	31,2	53,1	56,2	103,1	106,2	303,3
EPM9320LC84-15	320	56	2,8	17,8	5,3	32,1	10,3	60,7	117,6

Продовження таблиці 15.10

EPF6010ATC100-1	880	67	1	14,9	1,9	26,8	3,7	50,7	111,1
EPF8282ALC84-2	208	64	4,3	15,6	8,1	28,1	15,8	53,1	96,1
EPF10K30ETC144-1	1728	96	0,5	10,4	0,9%	18,7	1,9	35,4	250
			Розрядність						
Синхронний Радемахера	Число макро-комірок	Число входів	8		16		32		Частота МГц
			Макро-ком.	I\O %	Макро-ком. %	I\O %	Макро-ком. %	I\O %	
			10	10	24	18	64	34	
EPM3032ALC44-4	32	30	31,2	33,3	75	60	200	113,3	135,1
EPM7032LC44-6	32	32	31,2	31,2	75	56,2	200	106,2	178,5
EPM9320LC84-15	320	56	3,1	17,8	7,5	32,1	20	60,7	58,1
EPF6010ATC100-1	880	67	1,1	14,2	2,7	26,8	7,2	50,7	45,4
EPF8282ALC84-2	208	64	4,8	15,6	11,5	28,1	30,7	53,1	42,9
EPF10K30ETC144-1	1728	96	0,5	10,4	1,3	18,7	3,7	35,4	126,5

На базі даного дослідження виведено формули розрахунку максимальної кількості розрядів синхронного лічильника Галуа для певного типу ПЛІС.

Для різного типу ПЛІС (MAX3000, MAX7000, MAX9000, FLEX6000, FLEX8000, FLEX10K) згідно виразу

$$K_{\max} = \begin{cases} C, & \textit{при } R > C, \\ R-2, & \textit{при } \mathrm{R} < \mathrm{C}, \end{cases} \qquad (15.12)$$

де K_{max} – максимальна кількість розрядів лічильника; R, C – відповідно максимальна кількість виводів та кількість макрокомірок ПЛІС.

Дана формула дає можливість оптимально підібрати необхідний тип ПЛІС, шляхом розрахунку необхідної кількості макрокомірок та входо-виходів для реалізації синхронних лічильників з максимальною розрядністю згідно гістограми (рис. 15.47).

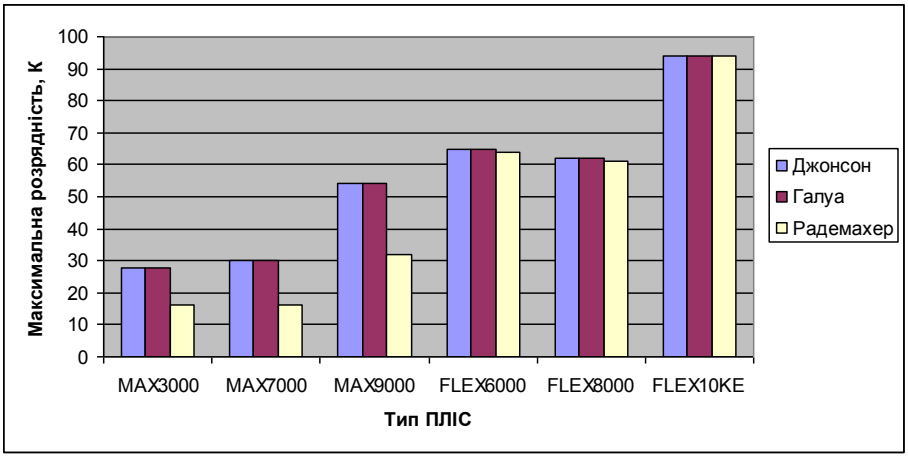


Рис. 15.47. Потенційні ресурси реалізації синхронних лічильників Джонсона, Радемахера, Галуа на кристалах ПЛІС різних типів.

Проведений аналіз показав, що ресурси кристалів ПЛІС меншої ємності, набагато ефективніше використовується при реалізації лічильників Джонсона та Галуа по відношенню до лічильників Радемахера. В той же час, для кристалів ПЛІС великої ємності ці переваги відсутні за рахунок більшої складності макрокомірок. Незважаючи на порівняно одинакову ефективність використання ресурсів ПЛІС різної ємності лічильники Галуа мають суттєву перевагу по відношенню до лічильників Джонсона, які характеризуються кодовою надлишковість базису Крейга відносно базису Галуа. Оскільки при однаковій розрядності К лічильники Джонсона можуть представити код максимального числа 2К, а лічильники Галуа 2^K-1. Таким чином лічильники Галуа характеризуються меншою кодовою розрядністю по відношенню до лічильників Джонсона, меншою структурною складністю та більш високою швидкодією до по відношенню до лічильників Радемахера, а в результаті найвищим рівнем ефективності використання ресурсів ПЛІС при заданій розрядності.

Отже, результати проведених досліджень дозволяють стверджувати, що лічильники Галуа володіють найвищими системними характеристиками і можуть бути ефективно використані для побудови спецпроцесорів з найвищими системними характеристиками по відношенню до існуючих.

На рис. 15.48 приведені оцінки використання площі ПЛІС фірми Altera (EPM9320LC84-15) при реалізації синхронних лічильників однакової кодової ємності в різних ТЧБ.

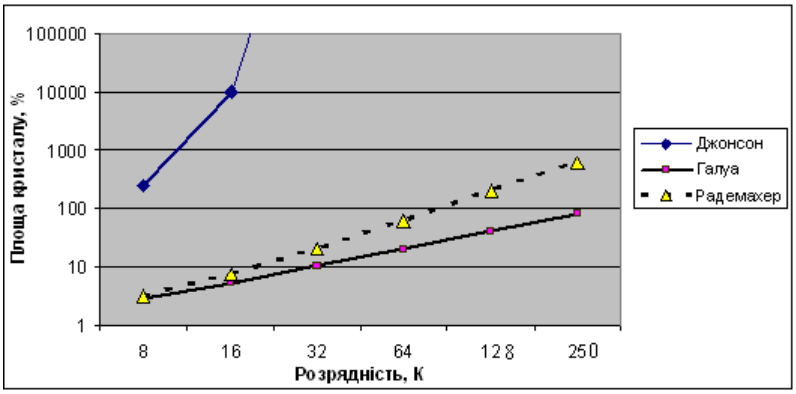


Рис.15.48. Графік залежності площі кристалу від розрядності лічильника для ПЛІС Altera.

З рис. 15.48 видно, що синхронні лічильники Галуа вимагають на порядок меншу кількість макрокомірок ніж лічильники базису Радемахера при однаковій розрядності. Це дозволяє зробити висновок, що лічильники Галуа при реалізації на ПЛІС забезпечують найменшу собівартість тиражування оскільки можуть бути реалізовані на кристалах меншої ємності та забезпечувати більш високу надійність, менші габарити, енергоспоживання та інше. В додатку Б приведені описи схем шифраторів, дешифраторів, суматорів, генераторів та лічильників у базисі Галуа, які реалізовані на мові VHDL.

15.13. Структура автономного сенсора з вихідними сигналами в кодах поля Галуа.

Структура розробленого одноканального АС показана на рис. 15.49

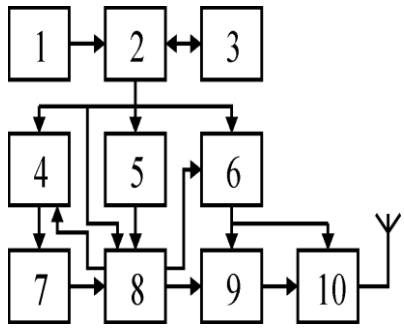


Рис. 15.49. Структурна схема автономного сенсора.

В даній схемі позначені такі функціональні блоки: 1 – альтернативне джерело живлення; 2 – блок живлення; 3 – акумулятор; 4, 6 – блоки

керування живленням; 5 – стабілізатор напруги; 7 – ПНЧ; 8 – мікроконтролер; 9 – спецпроцесор Галуа; 10 – передавач.

На рис. 15.50 представлено розроблену принципову електричну схему АС на основі одноканального СП Галуа.

Основні функції керування роботою АС виконує мікроконтролер 8 (типу Atmega8l). Оскільки АС повинен працювати в автономному режимі, тобто використовувати наявне автономне джерело живлення(АДЖ), то в залежності від області використання доцільно використовувати різні методи економії енергії, що дозволяє значно зменшити потужність АДЖ, а відповідно і габарити пристрою.

В даній схемі АС передбачені наступні роз'єми: X1 – призначений для під'єднання вхідного струмового сигналу (5-20 мА); до роз'єму X2 під'єднуються альтернативне джерело живлення та акумуляторна батарея; до роз'єму X3 під'єднується антена. Оскільки вихідна напруга джерела живлення може коливатися в певних межах (в залежності від освітленості, якщо використовується сонячна батарея, чи розр'яду акумулятора), то для забезпечення коректної роботи АС на його вузли подається стабілізована напругу живлення, для чого використовується 3 інтегральних стабілізатори DA1, DA2 та DA3 типу LM7803. Стабілізатор DA3 стабілізує напругу, яка подається на мікроконтролер, стабілізатори DA1, DA2 стабілізують напруги що подаються на ПНЧ та КШПС і радіопередавач відповідно. Причому передбачено окреме керування живленням ПНЧ та передавального модуля, що складається з ДШКП, який реалізований на ПЛМ DD3 типу EPM3032ATC44 та радіопередавача до складу якого входять наступні елементи: дросель D2; конденсатори C5-C11; резистори R8-R11, R13; варікап VD1 та транзистор VT1.

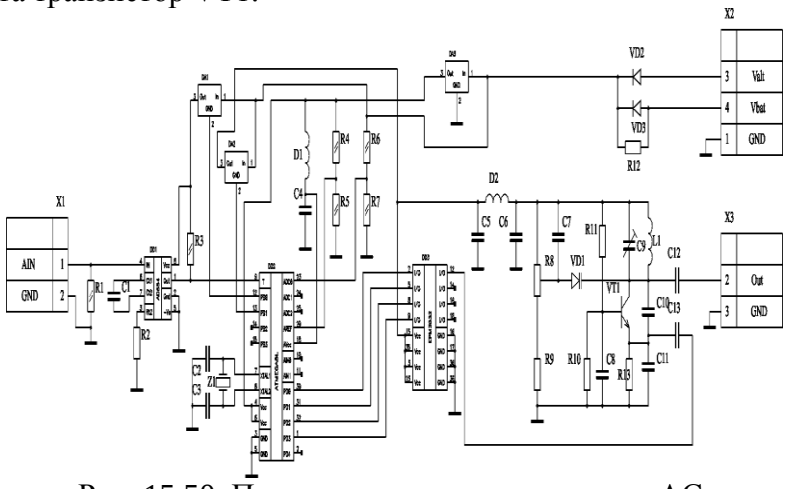


Рис. 15.50. Принципова електрична схема АС.

При роботі АС також необхідно контролювати напругу джерела живлення та при можливому відключені АС потрібно проінформувати про це комп'ютерну систему. Для цього через дільник напруги (R6, R7) на аналоговий вхід котролера подається напруга з джерела живлення який при її зниженні формує та передає код розряду батареї.

15.14. Структура модулів пам'яті колективного доступу на основі дешифраторів Галуа.

Пам'ять колективного доступу є важливим об'єднуючим структурним модулем RCG – процесора і виконує функції багатопортової пам'яті з колективним доступом.

Структура модуля пам'яті колективного доступу представлена на (рис.15.51)

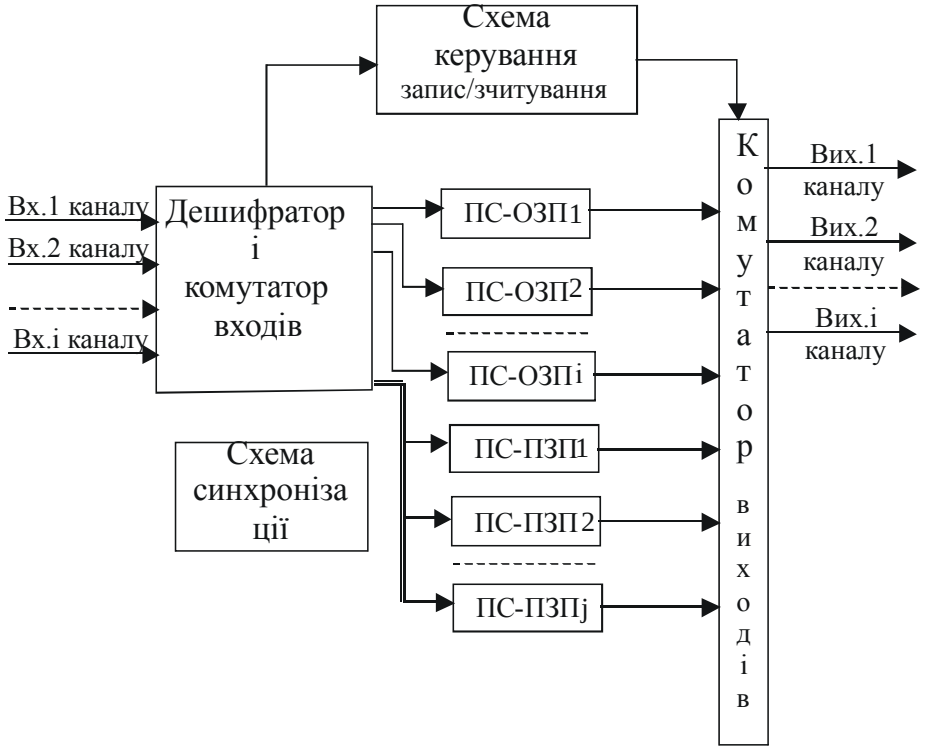


Рис. 15.51. Структурна схема пам'яті колективного доступу.

З рис.15.51 видно, що ПКД складається з таких основних функціональних вузлів:

- дешифратора і комутатора входів;
- комутатора виходів;

- схеми керування записом/зчитуванням;
- схеми синхронізації;
- "поштових скриньок" ОЗП і ПЗП.

Особлива властивість цього пристрою полягає в тому, що функціональні вузли рівномірно розподілені за однаковими блоками. Центральним модулем ПКД є модуль розпізнавання і комутації каналів. Модуль включає в себе логічні елементи "ВИКЛЮЧНЕ АБО", схеми “І”, "І-НЕ” і тактовані RS-тригери. На схему подаються сигнали попередньої установки (ПУ) , дозволи на зчитування і запис (Дзч, Дзап), а також імпульсні послідовності кодових посилок ключових слів для зчитування (Gзч.) і запису (Gзап), які створюються генератором Галуа. Робота пристрою відбувається синхронно за всіма входами, до яких підключені процесорні елементи (Вх1, Вх2, Вх3). Проте робота схеми відносно першого входу (Вх1) і будь-якого із входів (Вх2.Вх3) , а таких входів в конфігурації може бути і більша кількість, відрізняється.

Розглянемо роботу відносно другого входу. З генератора Галуа (Gзч) подається імпульсна послідовність, яка повинна забезпечити (це визначається кодом) працездатність каналу, що розглядається. Після імпульсу ПУ, який перекидає всі RS-тригери в одиничний стан, і подачі імпульсу дозволу зчитування, імпульс пакету даних, його адресна частина і ключ Галуа (Gзч) починають побітно порівнюватись на елементах "ВИКЛЮЧНЕ АБО". Якщо ключ Gзч відповідає каналу, що зчитується, відповідний RS-тригер не змінює свого стану (його прямий вихід в стані логічної "1"). В протилежному випадку, коли будь яка пара біт по одному з пакету даних Gзч не співпаде, поява логічної "1" на виході "ВИКЛЮЧНЕ АБО" приведе до перекидання відповідного тригера в протилежний стан, що забороняє передачу інформації крізь відповідні вихідні логічні елементи "І-НЕ". Таким чином, масиви інформації будуть зчитуватися тільки в тих каналах, в яких стан RS-тригерів не змінився. На ці ж самі вихідні елементи "І-НЕ" подається інформація з виходу комірок пам'яті оперативного чи постійного запам’ятовуючого пристрою. Така робота схеми відповідає режиму зчитування інформації.

Робота схеми відносно першого входу і виходу відрізняється від розглянутого режиму. Це пов'язано з тим, що по цьому каналу можливо здійснювати не тільки зчитування інформації аналогічно зчитуванню інших каналів, але і записувати дані, що розташовані в "інформаційній частині" пакету даних. Це здійснюється слідкуючим чином. Після першого етапу декодування, який повністю співпадає з процесом зчитування, як і в інших каналах, поступає сигнал на дозвіл запису. На Вх1 продовжує надходити пакет даних ("адресна частина", код операції) , а на вході Gзап з'являється імпульсний код у вигляді послідовності Галуа. На відповідному елементі "ВИКЛЮЧНЕ АБО" ці послідовності побітно порівнюються. Якщо код

запису (Gзап) співпадає з кодом операції "адресної частини" пакету даних, відповідний RS-тригер залишається в початковому стані. А це приводить до дозволу читання даних із відповідних комірок пам'яті. Таким чином, "код запису" - це умовна назва ключової послідовності. Іншими словами, для того, щоб по першому каналу здійснити зчитування інформації з комірок пам'яті, на вхід Gзап необхідно подавати єдину дозволену ключову комбінацію. Якщо ця комбінація не співпаде з кодом операції "адресної частини" пакету даних, відбувається запис інформації першого каналу в комірки пам'яті модуля розпізнавання і комутації каналів.

Таким чином, розглянута схема являє собою "поштову скриньку" для першого каналу. Тільки по цьому каналу можливий запис інформаційного пакету в комірки пам'яті. За всіма іншими каналами інформація може бути тільки прочитана.

З розглянутого режиму роботи першого каналу видно, що для нього найбільш сприятливі умови при запису інформації і обмежені умови при зчитуванні. Це пов'язано з тим, що зчитування даних із своєї “поштової скриньки” відбувається досить рідко. Частіше всього ця інформація вже знаходиться у абонента, який працює з першим каналом. В той же самий час, запис даних в "свою поштову скриньку" для забезпечення доступу до неї процесорних елементів, що працюють за другим чи третім каналами, найбільш доцільний.

При потребі, алгоритм роботи схеми, стосовно першого каналу, може бути змінений. Це можливо в тому випадку, коли в першому каналі частіше виконується зчитування даних із "своєї поштової скриньки".

В розглянутій структурі при записі інформації по шині Вх1 абонент, що записує, може побітно цю ж інформацію зчитувати. Для цього існує відповідний елемент "І-НЕ", який, маючи вихід по схемі з відкритим колектором, об'єднується з іншими виходами аналогічних елементів по шині Вих1. Такий режим роботи є зручним для контролю інформації, що записується в "поштову скриньку" (ПС).

Таким чином, запис інформації в ПС можливий від абонента, що підключений до першого каналу. Всі інші процесорні елементи можуть тільки зчитувати дані з даної ПС. Модуль розпізнавання і комутації каналів обслуговує одну "поштову скриньку». Принциповій схемі, робота якої розглядалася вище, умовно можна надати позначення "3/1". Це значить, що вона забезпечує зчитування інформації з ПС за трьома каналами, а запис в ПС за одним каналом. Для дослідження роботи такі модулі були поєднані в схему багатопортової ПКД, подану на рис. 15.52.

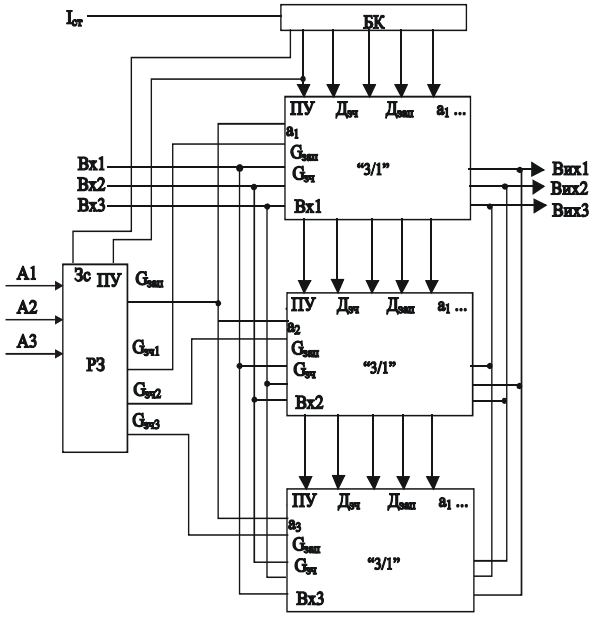


Рис. 15.52. Структурна схема пам’яті колективного доступу 3/1.

Ця схема являє собою пам'ять колективного доступу для трьох процесорних елементів (А1, А2, А3).

Модулі "3/1" підключені таким чином, що кожний з каналів може записувати інформацію в одну "свою" ПС, а зчитувати з ПС будь-якого каналу. Регістр зсуву (РЗ) являє собою генератор Галуа, який створює імпульсні кодові послідовності Gзч1-Gзч3 і Gзап. Перші три ключі використовуються для дешифрації номеру каналу, до якого звертається кожний з процесорних елементів. За ключем Gзап на другому етапі дешифрації адреси, класифікується вид операції, що виконуватиметься (запис/читання). Робота схеми здійснюється синхронно за стробуючими імпульсами (Іст.). Схема зрівноваження і синхронізації, здійснює формування циклу роботи, в який входять керуючі імпульси (ПУ, Дзч, Дзап, Зс) і імпульси, що створюють паралельний двійковий код a_1 - a_n для послідовного вибору кожної комірки пам'яті з "поштових скриньок". Вибір цих комірок здійснюється одночасно у всіх ПС.

Для дослідження можливості створення ПКД з більшою кількістю ПС, призначених для запису інформації від кожного абонента, розроблена схема ПКД з умовною назвою "З/З", яка подана на рисунку 15.53. Ця структура дозволяє будь-якому з трьох процесорних елементів записувати дані в "свої" три ПС, а зчитувати - з будь-якої (в даному випадку однієї з дев'яти). Виходячи з конструктивної доцільності, комірки пам'яті всіх ПС згруповані в окремий блок (ОЗП). Проте, керування цими ПС і способи

запису/зчитування інформації, в порівнянні з попередніми схемами, не змінювались. Модернізувався тільки генератор Галуа, який повинен генерувати дев'ять ключових слів, щоб забезпечити можливість звертання до всіх ПС.

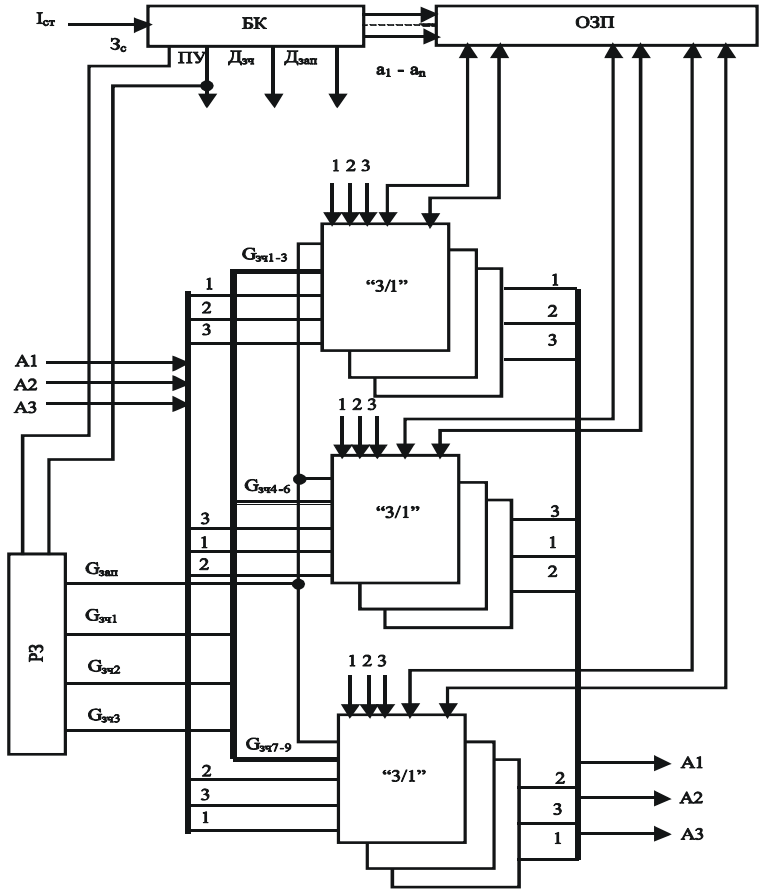


Рис. 15.53. ПКД для трьох пристроїв із структурою типу 3/3.

Реалізуючи розглянуті вище принципи, розроблена ПКД із модулями керування і комутації "8/1" (вісім процесорних елементів зчитують інформацію, один - записує) представлена на рис.15.54.

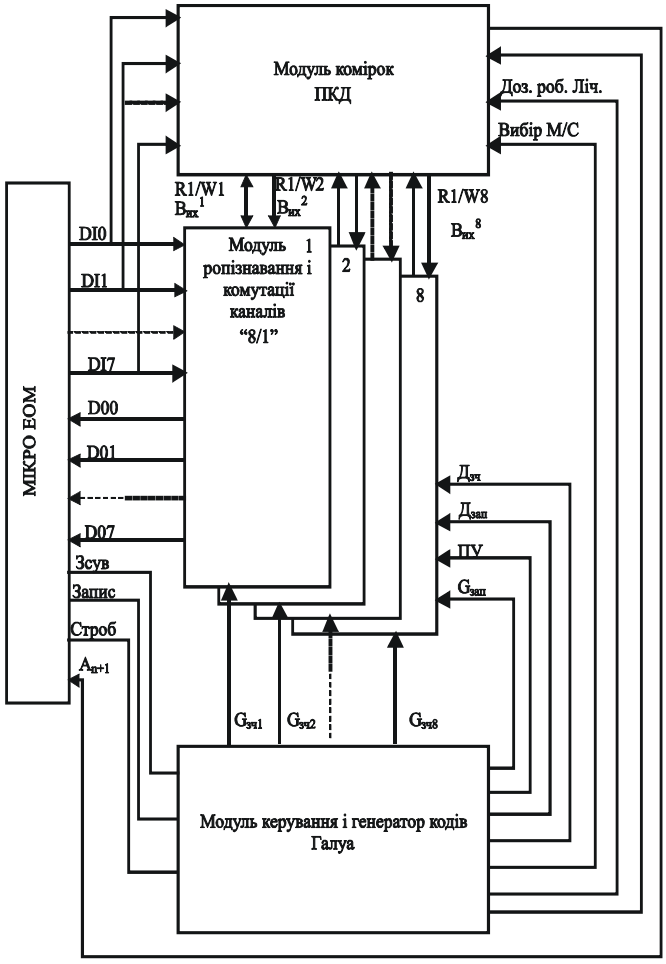


Рис. 15.54. Структурна схема 8- ми портової ПКД.

Така структура дозволяє провести дослідження працездатності пам'яті за допомогою одного восьмирозрядного комп’ютера. Кожний з розрядів цієї ЕОМ виконує функцію автономного, послідовно підключеного абонента. Використання паралельних портів для виводу і вводу інформації забезпечує синхронність роботи процесорних елементів. На кожний із розрядів, у вигляді послідовності імпульсів, подавався пакет даних відповідного умовно-незалежного абонента. Кожна з ПС складалася з 2048К комірок пам'яті.

Для аналізу запропонованої структури розглядалися параметри продуктивності і часу відповіді системи обробки інформації (СОІ) , які найбільш повно і точно визначають можливості системи при обробці відповідної кількості вхідних задач.

Продуктивність оцінюється кількістю задач, що вирішує система за одиницю часу:

$$\lambda = \frac{n}{t}, \tag{15.13}$$

де n - кількість задач, що обробляється системою за час t.

Якщо позначити інтенсивність вхідного потоку задач системи, як Λ, то залежність продуктивності системи від Λ, при врахуванні кількості вхідних задач, або послідовних інформаційних каналів для різних топологій процесорів, може бути розрахована згідно (15.13). На рис. 15.55 показано порівняння характеристик продуктивності у залежності від топології та інтенсивності процедур запису/читання. В області $0 \le \Lambda \le \lambda^x$ інтенсивність вихідного потоку повністю визначається інтенсивністю вхідного потоку: $\lambda = \Lambda$. При $\Lambda > \lambda^x$ система із-за обмеженості ресурсів (кількості і швидкодії пристроїв, а також ємності пам'яті) не може протягом одиниці часу обслуговувати всі надіслані на обробку завдання. В результаті інтенсивність вихідного потоку λ, досягнувши крайнього значення λ^x, надалі залишається постійною при будь-яких значеннях $\Lambda > \lambda^x$. Значення λ^x визначає максимальну продуктивність системи для заданого класу задач і є характеристикою самої системи, яка не залежить від інтенсивності вхідного потоку задач (рис. 4.7).

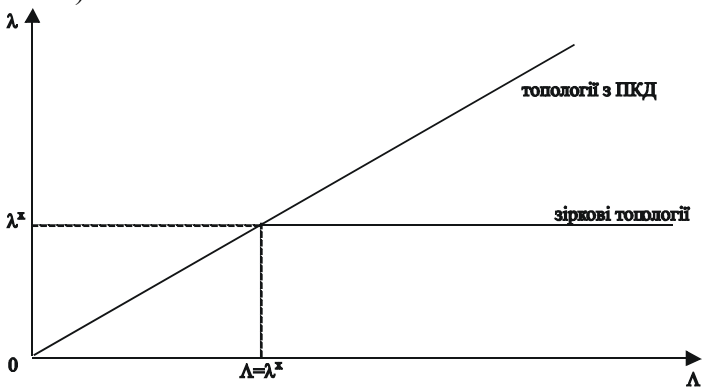


Рис. 15.55. Залежність продуктивності систем від інтенсивності вхідного потоку задач.

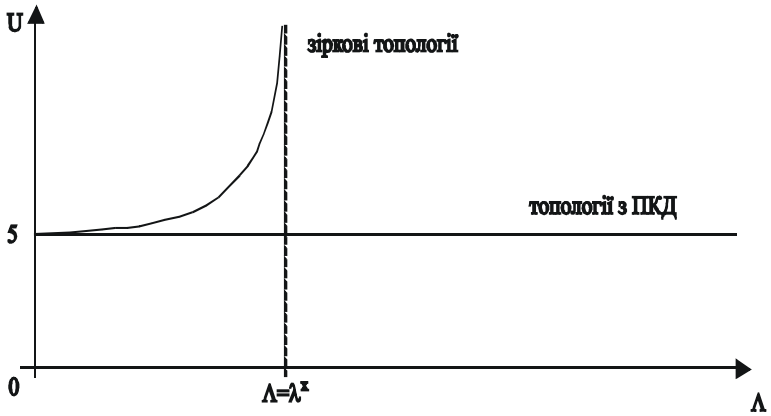


Рис. 15.56. Залежність часу відповіді системи від інтенсивності вхідного потоку задач.

Для запропонованої зірково-магістральної топології з ПКД такого обмеження не існує, що демонструють рис. 15.55. та рис.15.56.

До значення $\Lambda = \lambda^x$ вплив інтенсивності вхідного потоку задач на продуктивність аналогічний вище розглянутим системам. При значеннях $\Lambda \geq \lambda^x$ продуктивність системи продовжує зростати за тим же самим законом, в залежності від збільшення потоку вхідних задач. Враховуючи, що кількість вхідних задач для структури з ПКД аналогічна кількості вхідних каналів в ПКД, продуктивність такої системи зростатиме лінійно в усьому діапазоні $0 \leq \Lambda \leq \infty$. Таке явище пояснюється тим, що за рахунок схемної реалізації кожний абонент працює в "монопольному" режимі із ПКД, без конфліктів і зіткнень з іншими працюючими процесорними елементами.

Час відповіді процесора, як випадкова величина, найбільш повно характеризується функцією розподілу P (u<x) або функцією щільності імовірності p (u). Здебільшого час відповіді оцінюється середнім значенням, яке визначається як статистичне середнє випадкової величини u_i i=1, 2, ..., n для кожної поточної задачі

$$U = \frac{1}{2}\sum_{i=1}^{n} u_i \,. \qquad (15.14)$$

Час відповіді складається з двох частин: часу виконання задачі і часу очікування. Час виконання задачі при відсутності паралельних процесів дорівнює сумарній довжині всіх етапів процесу (вводу, звертання до зовнішньої пам'яті, процесорної обробки, виводу). Час виконання задачі залежить від складності обчислень Δ_1, Δ_2, ..., Δ_n і швидкодії V_1, V_2, …, V_n пристроїв 1, 2, ...N

$$\delta = \sum_{i=1}^{N} \Delta_i / V_i \,. \qquad (15.15)$$

Час очікування - це сума відрізків часу, на протязі яких задача, чи канал знаходилися в стані очікування необхідних ресурсів. Очікування виникає при мультипрограмній обробці, коли ресурс, який потрібний для вирішення однієї задачі, зайнятий іншою задачею, і перша задача не виконується, очікуючи звільнення ресурсу. Цей режим аналогічний режиму багатоканального мережевого доступу до загальносистемного ресурсу. Розглянемо залежність середнього часу відповіді процесора U від інтенсивності вхідного потоку задач Λ. При Λ>0 час відповіді U=δ, де δ - визначається з формули (15.15). Із збільшенням Λ середній час відповіді монотонно зростає і може приймати достатньо велике значення, якщо інтенсивність вхідного потоку перевищує продуктивність процесора λ^x ($\Lambda > \Lambda^x$) протягом великого періоду часу.

У зірковій топології з використанням ПКД при постійному зростанні кількості вхідних задач чи кількості вхідних процесорних елементів, що спілкуються з ПКД, час відповіді процесора не змінюється і завжди залишається постійним на рівні U = δ.

Таким чином, як показав аналіз, продуктивність і час відповіді системи в першу чергу залежать від структури і технічних характеристик загальносистемних ресурсів. Проте, на значення продуктивності і часу відповіді процесора не другорядним чином впливають засоби, що забезпечують доступ незалежних процесорних елементів до загальних ресурсів.

15.15. Процесорні модулі виконання арифметичних операцій в базисах Крестенсона, Галуа.

Основними операційними елементами RCG – процесора є модулі виконання арифметичних операцій, які являють собою незалежні спеціалізовані процесори призначені для швидкого виконання арифметичних операцій у відповідних теоретико-числових кодових базисах

Структурна схема арифметичного модуля в базисі Радемахера представлена на рис.15.57.

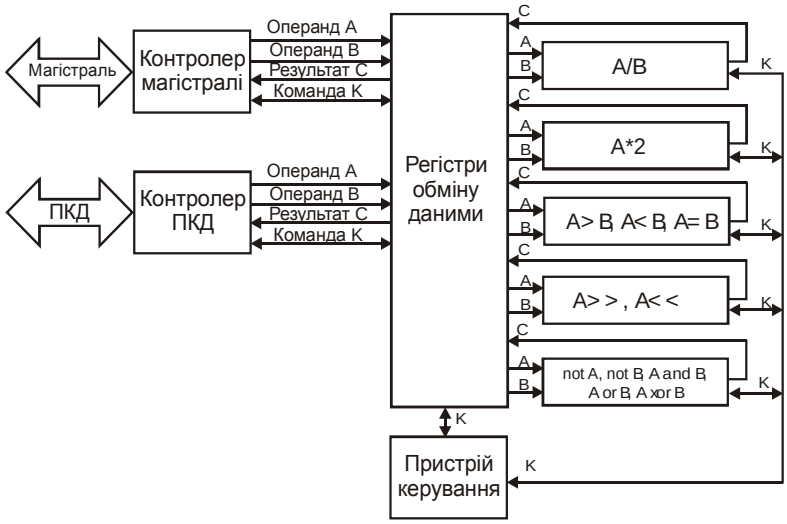


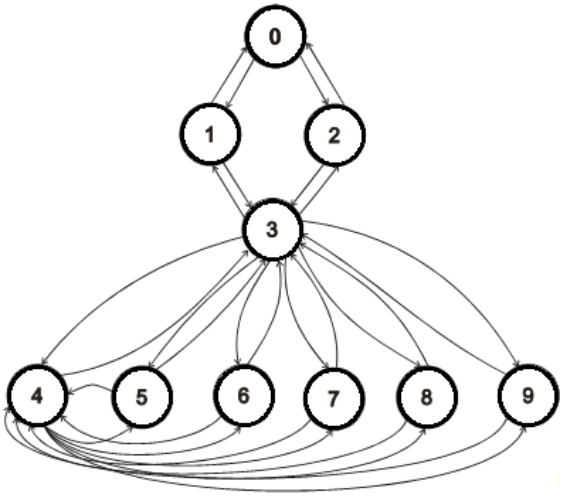
Рис. 15.57. Структурна схема арифметичного модуля в кодовому базисі Радемахера.

До складу арифметичного модуля в кодовому базисі Радемахера входять такі функціональні елементи:

- пристрій керування;
- контролер магістралі;
- контролер ПКД;
- регістри обміну даними;
- блоки виконання операцій ділення, множення на 2, порівняння, зсуву та логічних операцій;
- шини даних та команд.

Схема працює наступним чином: контролери ПКД та магістралі здійснюють моніторинг пакетів даних і при виявленні пакету, призначеного для арифметичного модуля Радемахера, здійснюють його декодування і запис операндів А , В та команди $К_і$ у відповідні регістри обміну даними. Пристрій контролю генерує сигнал активації для відповідного блоку виконання операцій і переходить у стан очікування.

Після виконання над операндами А і В блок виконання операцій заносить результат С у відповідний регістр та повідомляє пристрій керування про завершення операції. Пристрій керування записує результат С в буфер контролера ПКД або магістралі в залежності від вимоги. Приклад реалізації процесу роботи модуля можна представити у вигляді графа мікропрограми (рис.15.58.) :



0 – Ініціалізація арифметичного модуля;
1 – Контроль та обмін даними з ПКД;
2 – Контроль та обмін даними з магістраллю;
3 – Операції регістру обміну даними;
4 – Операції пристрою керування;
5 – Арифметичне ділення;
6 – Множення на 2;
7 – Операції порівняння;
8 – Операції зсувів;
9 – Логічні операції.

Рис. 15.58. Граф мікропрограми функціонування арифметичного модуля в базисі Радемахера.

Вершинам графу присвоєні номери 0, 1, …, 9, що відповідає кожному функціональному елементу, який виступає оператором модуля арифметичних операцій.

За допомогою даної графової моделі розраховуємо тривалість виконання мікропрограм для виконання операцій ділення, множення на 2, порівняння, зсуву та логічних операцій в базисі Радемахера.

Кожній вершині присвоюється T_i - час виконання такту мікрооперації, що дорівнює:

$$T_i = \tau_i + \vartheta_i, \tag{15.16}$$

де τ_i - проміжок часу на керування, ϑ_i - проміжок часу на виконання мікрооперації.

Якщо такт T_i має змінну тривалість, то вершина і позначається трьома можливими значеннями $(T_i = T_{min}, T_i = \overline{T}, T_i = max,)$.

Тривалість виконання мікропрограми визначається сумою тривалості тактів:

$$t = \sum_{i=1}^{m} n_i T_i, \tag{15.17}$$

де n_i - кількість звернень до і-го оператора.

Згідно (15.16) та (15.17) обчислимо час виконання мікропрограм для арифметичного модуля в базисі Радемахера (1 – процес ініціалізація арифметичного модуля, 2 – процес контролю та обміну даними з ПКД та

магістраллю, 3 – процес виконання операцій регістру обміну даними, 4 – процес виконання операцій пристрою керування, 5 – процес виконання арифметичних операцій) див. (рис. 15.59.).

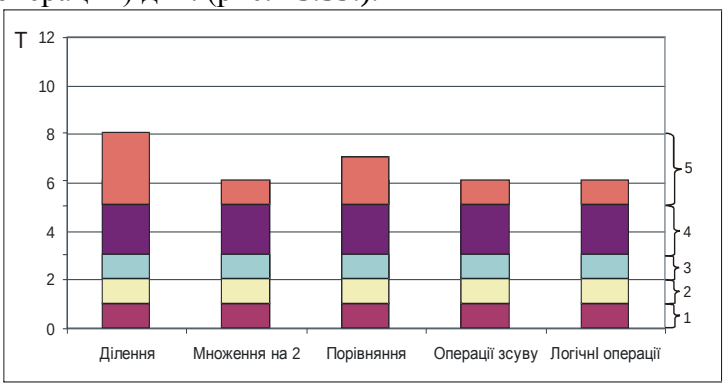


Рис. 15.59. Час виконання мікропрограм арифметичного модуля в базисі Радемахера.

З рис. 15.59 можна зробити висновок, що найменший час виконання мікропрограм є для функцій множення на 2, операцій зсуву та логічних операцій, а мікропрограма операції ділення займає найбільший час виконання.

Проведемо аналіз виконання мікропрограм для модуля в базисі Крестенсона. Даний модуль виконує тільки дві базові арифметичні операції додавання і множення, оскільки ці операції найефективніше і найпростіше реалізуються в СЗК.

Структурна схема арифметичного модуля в базисі Крестенсона представлена на рис.15.60.

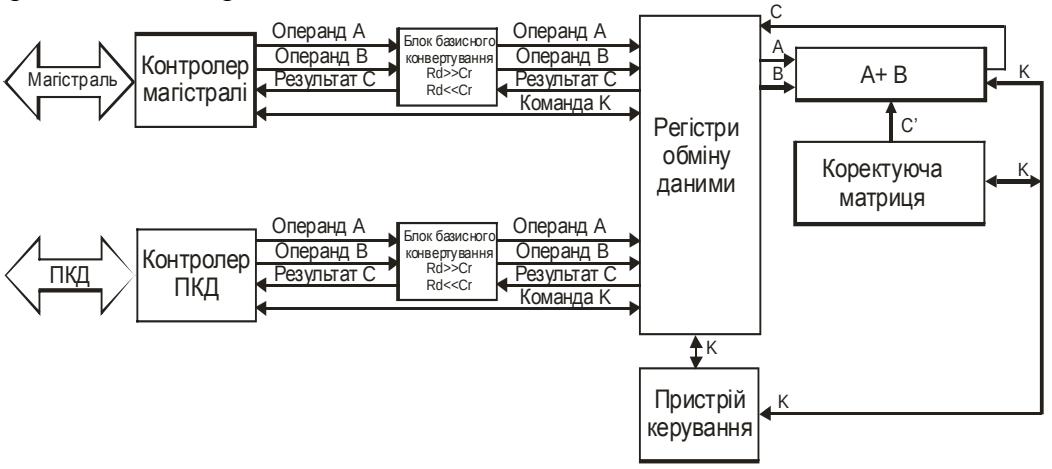


Рис. 15.60. Структурна схема арифметичного модуля в кодовому базисі Крестенсона.

До складу арифметичного модуля в кодовому базисі Крестенсона входять такі функціональні елементи:

– пристрій керування;

– контролер магістралі;

– контролер ПКД;

– блоки базисного перетворення;

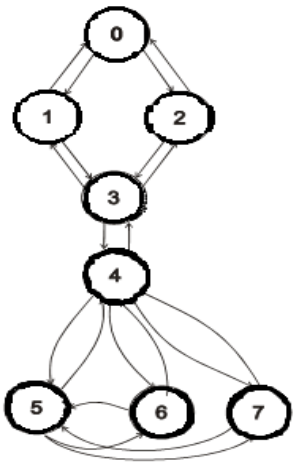
– регістри обміну даними;

– блоки виконання операцій сумування та коректуюча матриця множення;

– шини даних та команд.

Схема працює наступним чином: контролери ПКД та магістралі здійснюють моніторинг пакетів даних і при виявленні пакету, призначеного для арифметичного модуля Крестенсона, здійснюють його декодування, переведення операндів A і B за допомогою блоку базисного перетворення в двійкове представлення базису Крестенсона і запис операндів A, B та команди K_i у відповідні регістри обміну даними. Пристрій контролю генерує сигнал активації для відповідного блоку виконання операцій і переходить у стан очікування.

Після виконання над операндами A і B блок виконання операцій заносить результат C у відповідний регістр та повідомляє пристрій керування про завершення операції. Пристрій керування передає результат C на блок базисного перетворення, після якого C в базисі Радемахера надходить у буфер контролера ПКД або магістралі в залежності від вимоги. Приклад реалізації процесу роботи модуля можна представити у вигляді графа мікропрограми (рис. 15.61.) :



0 – Ініціалізація арифметичного модуля;
1 – Контроль та обмін даними з ПКД;
2 – Контроль та обмін даними з магістраллю;
3 – Базисні перетворення;
4 – Операції регістру обміну даними;
5 – Операції пристрою керування;
6 – Сумування;
7 – Множення.

Рис. 15.61. Граф мікропрограми функціонування арифметичного модуля в базисі Крестенсона.

Вершинам графу присвоєні номери 0, 1, …, 7, що відповідає кожному функціональному елементу, який виступає оператором модуля арифметичних операцій.

За допомогою даної графової моделі розраховуємо тривалість виконання мікропрограм для рішення операцій Сумування та множення.

Згідно (4.4) та (4.5) обчислимо час виконання мікропрограм для арифметичного модуля в базисі Крестенсона (1 – процес ініціалізація арифметичного модуля, 2 – процес контролю та обміну даними з ПКД та магістраллю, 3 – процес міжбазисного перетворення, 4 – процес виконання операцій регістру обміну даними, 5 – процес виконання операцій пристрою керування, 6 – процес виконання арифметичних операцій) див. (рис. 15.62.).

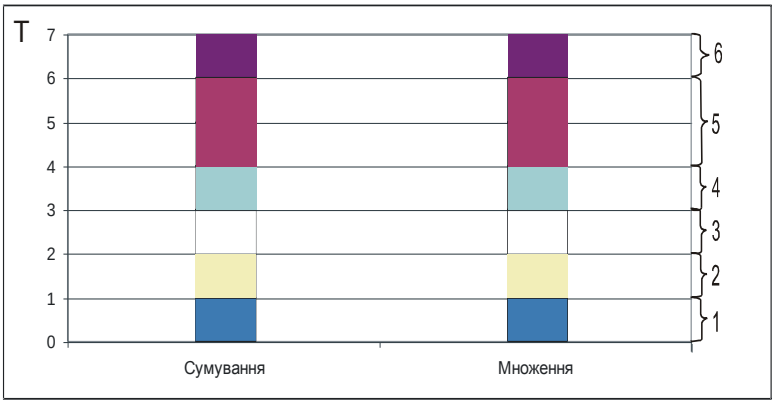


Рис. 15.62. Час виконання мікропрограм арифметичного модуля в базисі Крестенсона.

З рис. 15.62 можна зробити висновок, що час виконання мікропрограм для функцій множення та сумування є однаковим, що дозволяє реалізовувати складні операції для задач ЦОД.

Проведемо аналіз виконання мікропрограм для модуля в базисі Галуа, який реалізує арифметичні операції інкрементування і декрементування, що реалізуються шляхом зсуву регістру Галуа вправо або вліво.

Структурна схема арифметичного модуля в базисі Галуа представлена на рис.15.63.

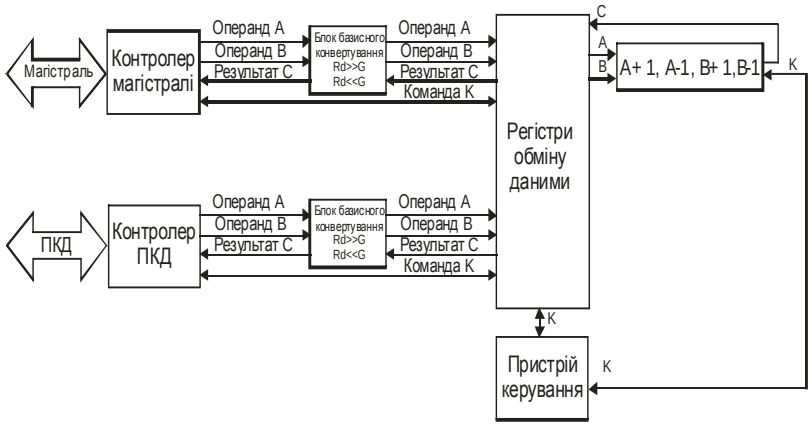


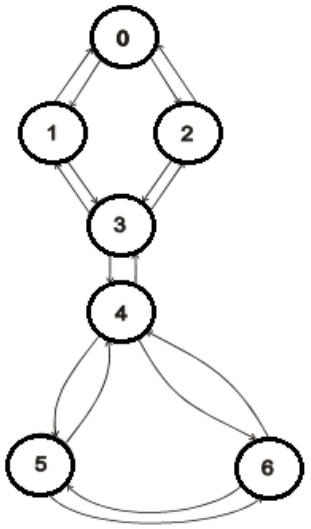
Рис. 15.63. Структурна схема арифметичного модуля в кодовому базисі Галуа.

До складу арифметичного модуля в кодовому базисі Галуа входять такі функціональні елементи:

- пристрій керування;
- контролер магістралі;
- контролер ПКД;
- блоки базисного перетворення;
- регістри обміну даними;
- блоки виконання операцій інкрементування та декрементування;
- шини даних та команд.

Схема працює наступним чином: контролери ПКД та магістралі здійснюють моніторинг пакетів даних і при виявленні пакету, призначеного для арифметичного модуля Галуа, здійснюють його декодування, переведення операндів А і В за допомогою блоку базисного перетворення в двійкове представлення базису Галуа і запис операндів А або В та команди $К_i$ у відповідні регістри обміну даними. Пристрій контролю генерує сигнал активації для відповідного блоку виконання операцій і переходить у стан очікування.

Після виконання над операндами А або В блок виконання операцій заносить результат С у відповідний регістр та повідомляє пристрій керування про завершення операції. Пристрій керування передає результат С на блок базисного перетворення, після якого С в базисі Радемахера надходить у буфер контролера ПКД або магістралі в залежності від вимоги. Приклад реалізації процесу роботи модуля можна представити у вигляді графа мікропрограми (рис.15.64.) :



0 – Ініціалізація арифметичного модуля;
1 – Контроль та обмін даними з ПКД;
2 – Контроль та обмін даними з магістраллю;
3 – Базисні перетворення;
4 – Операції регістру обміну даними;
5 – Операції пристрою керування;
6 – Інкрементування / декрементування;

Рис. 15.64 Граф мікропрограми функціонування арифметичного модуля в базисі Галуа.

Вершинам графу присвоєні номери 0, 1, …, 6, що відповідає кожному функціональному елементу, який виступає оператором модуля арифметичних операцій.

За допомогою даної графової моделі розраховуємо тривалість виконання мікропрограм для рішення операцій Сумування та множення.

Згідно (15.17) та (15.18) обчислимо час виконання мікропрограм для арифметичного модуля в базисі Галуа (1 – процес ініціалізація арифметичного модуля, 2 – процес контролю та обміну даними з ПКД та магістраллю, 3 – процес міжбазисного перетворення, 4 – процес виконання операцій регістру обміну даними, 5 – процес виконання операцій пристрою керування, 6 – процес виконання арифметичних операцій інкрементування / декрементування) див. (рис. 15.65).

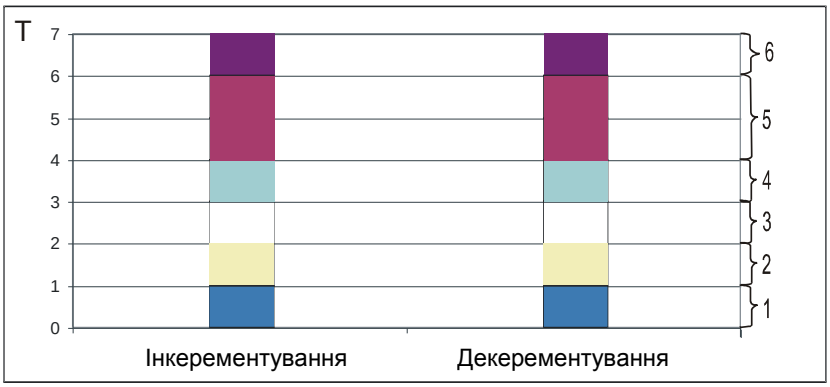


Рис. 15.65 Час виконання мікропрограм арифметичного модуля в базисі Галуа.

З рис. 15.65. можна зробити висновок, що час виконання мікропрограм для функцій інкрементування та декрементування виконується за один системний такт, і не потребує складного апаратного забезпечення і дозволяє дозволяє реалізовувати складні операції для задач ЦОД.

15.16. Схемотехнічна реалізація модулів пам’яті колективного доступу.

Основою RCG – процесора з зірково-магістральною топологією є модуль контролера каналів абонентів, що підключаються до ПКД. Найбільш доцільною структурою модуля є структура, яка забезпечує зчитування інформації із ПС всіма абонентами, а запис – одного абоненту, що підключений до нульового каналу. Такий модуль структури “8/1” обслуговує одну ПС. При потребі збільшення кількості ПС, призначених для запису інформаційних пакетів певного абонента, модуль “8/1” поєднується з аналогічними для організації необхідної структури ПКД (8/1; 8/2; ...8/n). Спроектований модуль дозволяє нарощувати можливості пам’яті загального користування не тільки по режиму запису, але і по режиму зчитування інформації з даної ПС декількома абонентами, кількість яких перевищує вісім.

Схема спроектована за принципами, що розглянуті в підрозділі 4.1. Модуль складається із одної ПЛІС серії Spartan-3. Логічні елементи мають відкриті колекторні виходи, що дозволяє об'єднувати їх з аналогічними

виводами інших модулів за схемою технологічного "АБО". Швидкодія всієї ПКД залежить від елементної бази, що застосована в комірках пам'яті ПС.

На базі розглянутого модуля контролера каналів розроблена загальносистемна пам'ять загального користування, яка є центральним пристроєм радіальної мережі. Дана ПКД обслуговує до 64 абонентних станцій, кожна з яких записує інформацію в "свою" ПС, а зчитує - з будь - якої ПС. Крім цього, в ПКД закладено 190 ПС, що виконують функцію ПЗП. В цих комірках в пакетах довжиною 2 Мбіти знаходяться фрагменти сервісних програм, які часто використовуються користувачами. Фрагменти, по потребі, збираються абонентними станціями, причому цей процес не заважає іншим станціям спілкуватися з відповідними поштовими скриньками.

Експлуатація розробленої ПКД підтверджує гнучкість модуля контролерів, при розширенні функціональних можливостей і створенні структур зірково-магістральної топології. Модуль контролера архітектури "8/1" має 24 контактних з'єднання (з врахуванням загальної шини і живлення). Модернізація модуля полягає в розробці мікросхеми, що повністю виконує функції схеми "8/1", і виготовлення базоматричного кристалу з функціями восьми модулів контролерів пам'яті загального користування.

Розроблена схема керування ПКД, створена на базі цифрових логічних елементів, забезпечує взаємодію всіх складових частин як самої пам'яті, так і всієї зірково - магістральної мережі. Цей пристрій виконує наступні функції:

- генерує сигнали "попередньої установки", дозволу "зчитування" і "запису" для роботи модулів контролера каналів ПКД;

- керує роботою комірок, пам'яті ПС за допомогою сигналів вибірки мікросхем пам'яті (CS) і дозволу роботи лічильників адреси;

- відтворює еталонні ключові кодові послідовності Галуа, які використовуються для зчитування і запису інформації в ПС;

- генерує синхронізуючу послідовність (S) , що передається кожному абоненту для визначення початку робочого циклу.

Використання цифрових логічних елементів в цьому пристрої обумовлено обмеженою кількістю функцій, які виконує автомат, а також відсутністю потреби в зміні алгоритму роботи системи в процесі експлуатації всієї системи. Враховуючи ці основні фактори, використання елементів мікропроцесорної техніки є недоцільним.

Робота пристрою циклічна. Кожний цикл починається з імпульсу синхронізації початку, за яким, під час появи наступного стробуючого імпульсу, пропускається один синхронізуючий. Далі іде пачка імпульсів, кількість яких дорівнює кількості комірок пам'яті в ПС. В створеній ПКД вибрана глибина пакету ПС в 2048 кбіт.

Принципова схема цифрового автомату складається з двох блоків: генераторного і керуючого. В генераторний блок входять: кварцовий генератор з опорною частотою 100 мГц, лічильники, що формують довжину циклу роботи схеми, одновібратори для створення 4 мс імпульсів стробування і зсуву інформації. Головним сигналом, який формує генераторний блок, є синхронізуюча імпульсна послідовність S, що подається на кожну абонентну станцію і використовується останньою для визначення початку циклу роботи цифрового автомату, а відповідно, і початку циклу спілкування з вибраною ПС.

Керуючий блок за сигналами "Зсув", "Строб" і "Запис/попередня установка" формує сигнали, які подаються в модулі контролерів пам'яті і в модуль комірок поштових скриньок. Формування еталонних ключових комбінацій кодів Галуа відбувається в цьому блоці на регістрах зсуву. На початку циклу, в режимі паралельного завантаження, в розряди регістру записується початковий код Галуа. Синхронно із стробуючими імпульсами, переведений в режим послідовного зсуву інформації, регістр на відповідних виходах формує еталонні ключі Галуа, які і використовуються для дешифрації операцій зчитування і запису.

Особливість цього пристрою полягає в тому, що більшість функціональних вузлів рівномірно розподілені за однаковими блоками. Центральним модулем ПКД є модуль розпізнавання і комутації каналів, схема якого приведена на рис.15.66.

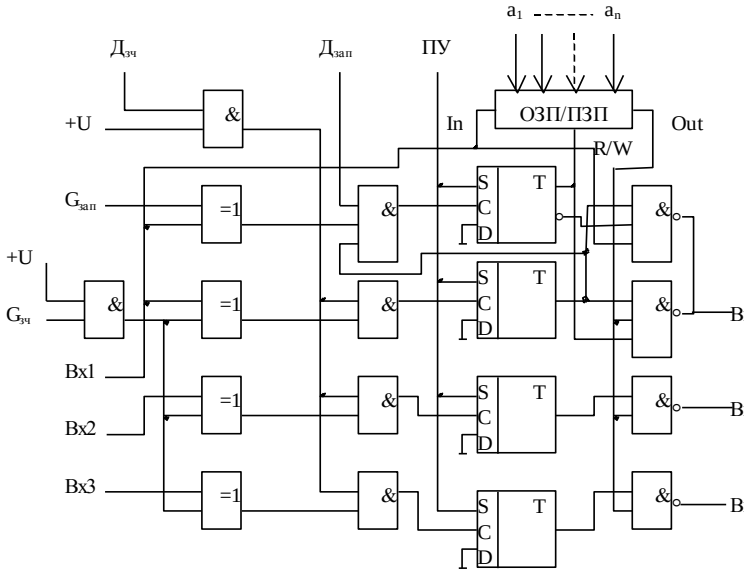


Рис.15.66. Схема модуля розпізнавання і комутації каналів.

На схему подаються сигнали попередньої установки (ПУ) , дозволи на зчитування і запис ($Д_{зч}$, $Д_{зап}$) , а також імпульсні послідовності кодових посилок ключових слів для зчитування ($G_{зч}$) і запису ($G_{зап}$) , які створюються генератором Галуа. Робота пристрою відбувається синхронно за всіма входами, до яких підключені абоненти.

Спосіб доступу для запису і зчитування пакету даних не залежить від типу ПС і є універсальним. Інформація про наміри абонента знаходиться в адресній частині пакету, що передається від абонента до ПКД (рис.15.67).

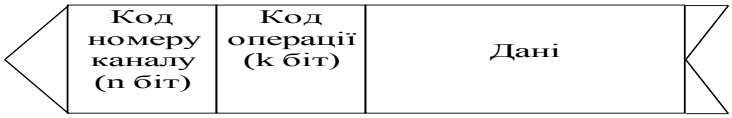


Рис.15.67. Формат пакету даних.

Блок даних складається з адресної та інформаційної частин. Адресна частина складається з n+k бітів. Перші n біт визначають номер ПС, з якою встановлюється зв'язок. Інші k біт використовуються як ключ, за яким визначають тип операції, що буде зараз виконуватись. Синхронізація одночасної роботи всіх елементів здійснюється синхроімпульсами, які передаються спеціальною синхронізуючою шиною до всіх абонентів одночасно. Частота синхронізуючої послідовності відповідає мінімально можливому часу здійснення операції зчитування/запису елементної бази.

З початком спілкування поступає "n" біт, які є адресом ПС, далі наступні "k" біт, що визначають тип операції, і після цього "m" біт – інформаційна частина. З цього можна визначити час доступу до системних ресурсів:

$$T_{доступу}=t_n+t_k+t_m, \qquad (15.19)$$

де t_n+t_k – час адресної частини,

t_m – час інформаційної частини пакету даних.

Запропонований метод доступу успішно може використовуватись не тільки в радіальних мережах, але і в шинних конфігураціях, як локальна підсистема, також в багатопроцесорних структурах, як універсальний компонент, з можливістю модульного нарощення собі подібних.

Це дає можливість паралельно звертатись до даних будь-якого функціонального елементу RCG - процесора без колізій та черг, що призводить до росту ефективності та працездатності системи.

15.17. Структура адресного дешифратора ПКД на ПЛІС.

При реалізації адресного дешифратора ПКД на ПЛІС було проведено дослідження його структури, яке показало, що даний тип дешифраторів має

регулярну структуру і складається з однотипних блоків. На основі однотипного блоку було спроектовано базовий примітив (рис.15.68).

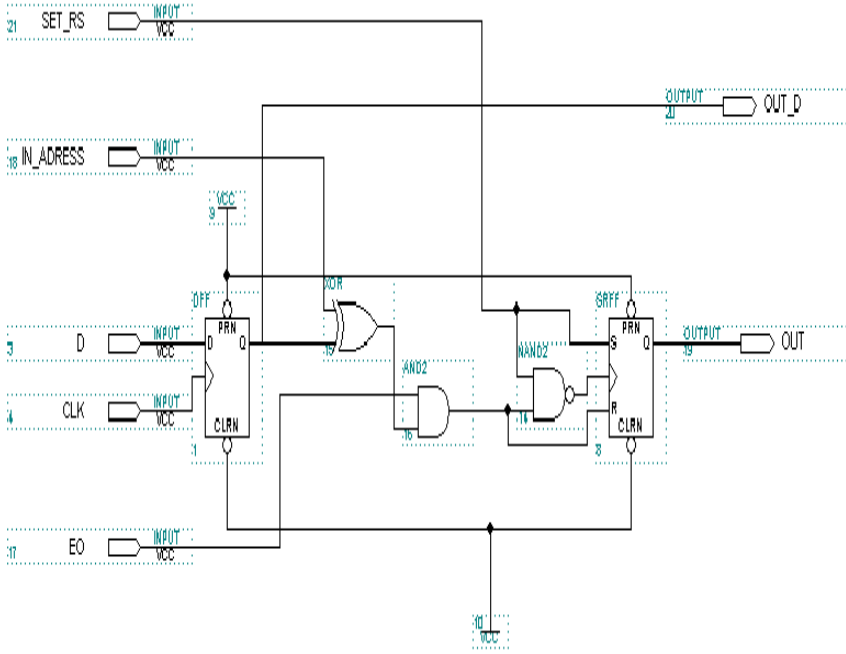


Рис. 15.68. Базовий примітив адресного дешифратора ПКД.
SET_RS – вхід встановлення RS тригера в одиничний стан; IN_ADRESS – вхід адресної послідовності каналу; D – інформаційний вхід на тригер регістру зсуву; CLK – вхід синхронізації; EO – вхід дозволяючого сигналу; OUT_D – вихід тригера регістру зсуву; OUT – вихід і- го каналую.

Використовуючи базовий примітив, який являє собою повноцінний одно-канальний адресний дешифратор Галуа здійснюємо розширення числа каналів шляхом збільшення кількості елементів (рис.15.69).

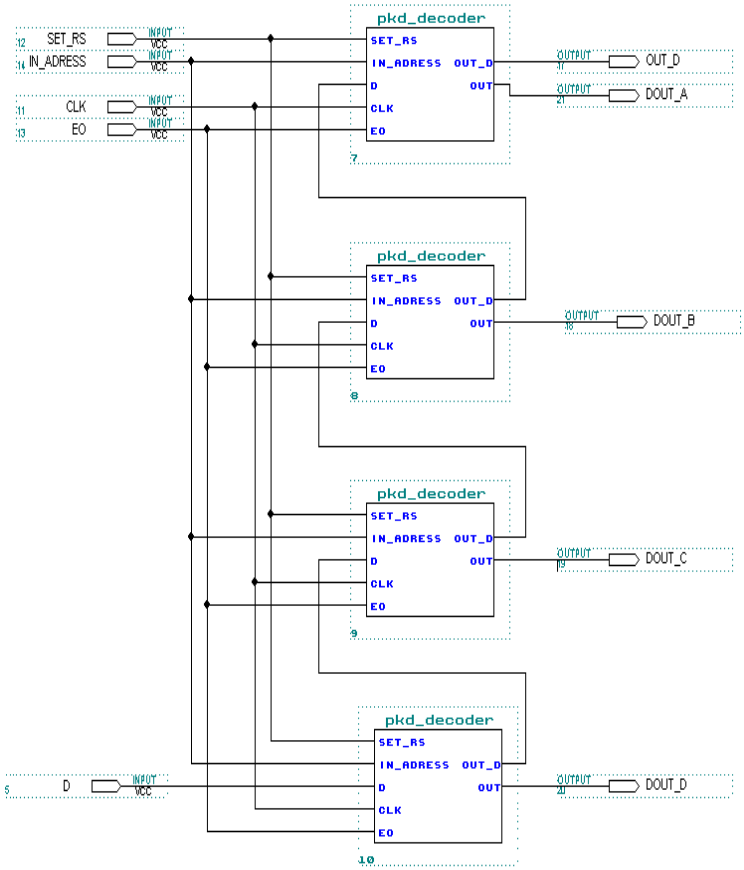


Рис 15.69. Реалізація 4-х канального адресного дешифратора Галуа. SET_RS – вхід встановлення RS тригера в одиничний стан; IN_ADRESS – вхід адресної послідовності каналу; D – інформаційний вхід на тригер регістру зсуву; CLK – вхід синхронізації; EO – вхід дозволяючого сигналу; OUT_D – вихід тригера регістру зсуву; DOUT_A-DOUT_D– виходи і- го каналу.

Як видно з рис.15.69, при збільшенні кількості каналів адресного дешифратора Галуа зростає лише кількість канальних виходів (DOUT_A-DOUT_D) , а число службових входів залишається сталим.

При реалізації адресного дешифратора Галуа на ПЛІС фірми Altera було проведено дослідження кількості макрокомірок та виводів, що використовуються при реалізації дешифраторів з різним числом каналів для різних сімейств ПЛІС.

В результаті досліджень було встановлено, що для сімейств MAX3000, MAX7000, MAX9000 один примітив адресного дешифратора використовує дві макрокомірки та 7 виводів, а для сімейств FLEX6000, FLEX8000, FLEX10K, ACEX1K відповідно 4 макрокомірки та 7 виводів.

При збільшенні кількості каналів збільшення кількості макрокомірок і виводів збільшується лінійно. На базі даного дослідження було виведено формули розрахунку максимальної кількості каналів адресного дешифратора Галуа для певного типу ПЛІС.

Для MAX3000, MAX7000, MAX9000:

$$K_{кан} = C/2, \quad (15.20)$$

при $R \geq C + 6$ кількість каналів;

$$K_{кан} = (R-6)/2, \quad (15.21)$$

при $R < C + 6$ кількість каналів;

Для FLEX6000, FLEX8000, FLEX10K, ACEX1K:

$$K_{кан} = C/4, \quad (15.22)$$

при $R \geq C + 6$ кількість каналів;

$$K_{кан} = (R-6)/4, \quad (15.23)$$

при $R < C + 6$ кількість каналів,

де $К_{кан}$ – кількість каналів,

R – максимальна кількість виводів ПЛІС що програмуються,

C – кількість макрокомірок ПЛІС.

Дані формули дають можливість оптимально підібрати необхідну ПЛІС, розрахувавши попередньо її ємність для реалізації адресного дешифратора Галуа (рис.15.70, рис.15.71).

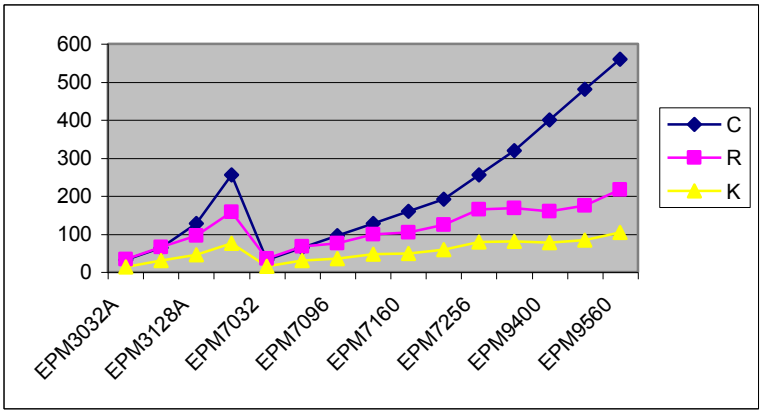


Рис. 15.70. Реалізація адресного дешифратора Галуа на ПЛІС MAX3000, MAX7000, MAX9000.
C – кількість макрокомірок, R – кількість програмованих входів, К – кількість каналів дешифратора Галуа.

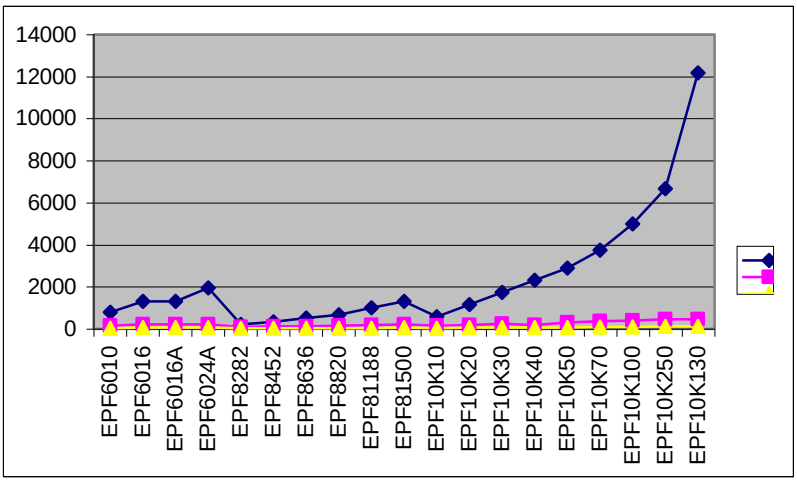


Рис. 15.71. Реалізація адресного дешифратора Галуа на ПЛІС серій FLEX6000, FLEX8000, FLEX10K, ACEX1K.

З рис.15.70.-15.71. коефіцієнт ефективності кристалів ПЛІС представлених типів розраховується за відповідними виразами:

$$E_{ефект} = \frac{\overline{\frac{C}{2}}}{K}, \qquad (15.24)$$

для серій MAX3000, MAX7000, MAX9000;

$$E_{ефект} = \frac{\overline{\frac{C}{4}}}{K}, \qquad (15.24)$$

для серій FLEX6000, FLEX8000, FLEX10K.

Результати розрахунку ефективності реалізації дешифратора Галуа за виразами (15.24) та (15.24) представлено на рис. 5.7.

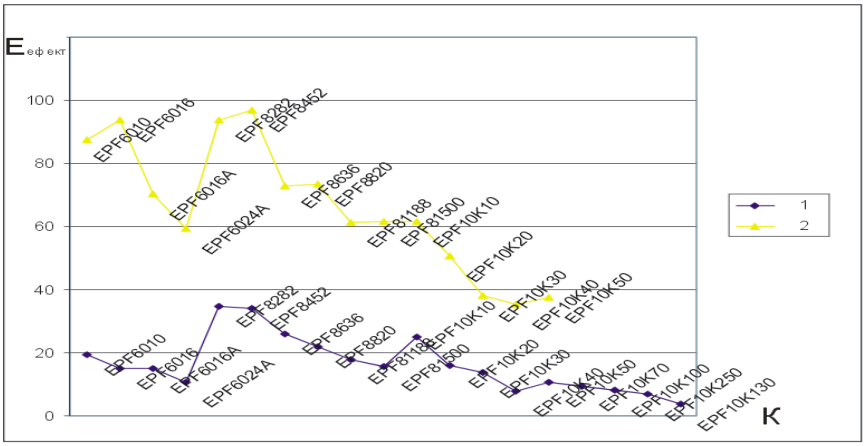


Рис. 15.72. Коефіцієнт ефективності при реалізація адресного дешифратора Галуа на ПЛІС :
1 – серії FLEX6000, FLEX8000, FLEX10K, ACEX1K; 2 – серії MAX3000, MAX7000, MAX9000; $E_{ефект}$ – коефіцієнт ефективності; К – кількість каналів дешифратора Галуа.

Проведені дослідження при реалізації адресного дешифратора Галуа на різних сімействах ПЛІС показали простоту реалізації даного типу елементів цифрової техніки. Завдяки регулярній структурі даний тип дешифраторів легко змінює кількість каналів адресації, що відкриває перспективу подальшого дослідження та розвитку елементів даного типу.

РОЗДІЛ 16

БАЗИ ДАНИХ В КОДАХ ГАЛУА

16.1. Організація лінійно-рекурентних баз даних в кодах поля Галуа.

Порівняння базисів Радемахера та Галуа показує, що коди матриці базису Радемахера, які забезпечують горизонтально-паралельне кодування даних відповідно відображається у вигляді вертикального вектора рекурентних кодів базису Галуа. При цьому вертикальна упаковка інформації є найбільш досконалою і компактною, дозволяє значно скоротити об'єм даних, які зберігаються в БД. Для того, щоб уникнути надлишковості кодів використовують кодову послідовність Галуа, яка базується на основі рекурентних кодонів Радемахера. Принципи та аналіз системних характеристик реалізації БД у базисі Галуа викладені у сумісних та особистих роботах автора та В.В.Шаряка.

Ієрархічна архітектура БД в базисі Галуа може бути представлена новою лінійно-рекурентною архітектурою (рис.16.1).

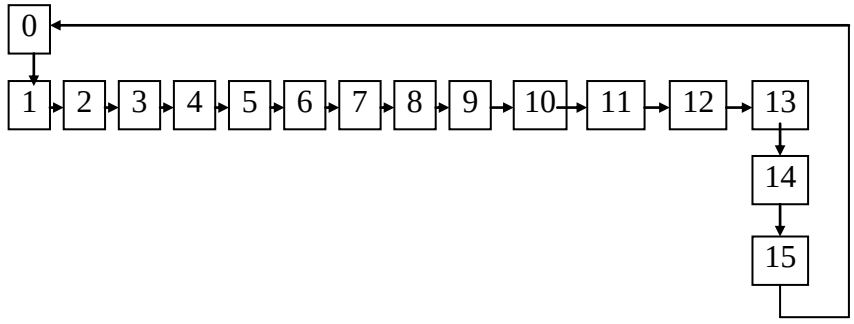


Рис.16.1. Лінійно-рекурентна архітектура БД в базисі Галуа.

Властивість рекурентності зв'зків ієрархічної БД при кодуванні елементів її в базисі Галуа дозволяє оптимізувати характеристики організації таких БД.

На основі петлі однорівневого циклу однорівневої схеми та вертикального вектора базиса Галуа можна побудувати вертикальні цикли з ланцюгами нової вертикально-рекурентної моделі БД в базисі Галуа. Вертикально-рекурентна модель бази даних в базисі Галуа складається з вертикальних відвітвлень послідовності ключових вузлів 0,1,2,…,n, які дублюються по принципу ,,Ханойської башти''. При цьому для кодування елементів ієрархічної моделі БД в базисі Радемахера необхідно дані

кодувати згідно з кодовою матрицею міжбазисного переходу. Очевидно, що даній кодовій матриці базису Радемахера однозначно відповідає матриця-вектор базису Галуа. При застосуванні базису Галуа реалізується рекурентний метод кодування елементів унітарного дерева. Ієрархічна модель БД в базисі Галуа може бути логічно представленою новою вертикально - рекурентною моделлю БД в базисі Галуа-вектор (рис.16.2) та вертикально - рекурентною моделлю БД в базисі Галуа з відвітвленням (рис.16.3).

Моделювання логічної структури однорівневої і дворівневої вертикально-рекурентної моделі БД в базисі Галуа:

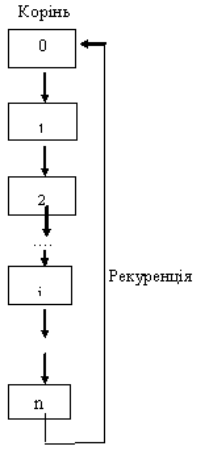


Рис. 16.2. Цикли однорівневої вертикально- рекурентної моделі БД Галуа з відвітвленням, де G_0- бітив базисі Галуа-вектор інвертованої кодовоїпослідовності Галуа.

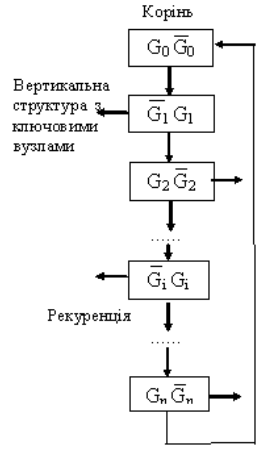


Рис. 16.3. Цикли дворівневої вертикально-рекурентної моделі БД в базисі Галуа з відвітвленням, послідовності Галуа, $\overline{G}_0$ - біти інвертованої кодової послідовності Галуа.

Особливістю кодування ідентифікаційних даних вертикально-рекурентної моделі БД в базисі Галуа є використання двобітового кодування ідентифікаторів даних з інверсіями. При цьому, якщо інвертується перший біт кодової послідовності Галуа, то розгалуження унітарного дерева відбувається схематично вліво або вниз , а при інвертуванні другого біта Галуа розгалуження відбувається вправо або вверх. Загальний приклад унітарного дерева з розгалуженнями вертикально-рекурентної моделі БД в базисі Галуа, показано на рисунку 5 на прикладі структури потоків даних корпорації „ Pallada Travell USA’’.

16.2. Організація ієрархічних баз даних в кодах поля Галуа.

Очевидно, що застосування базисів з високою надлишковістю кодових матриць (наприклад: Унітарна, Хаара, Крейга) є неефективними при кодування інформації в базах даних.

При використанні базису Радемахера для кодування даних БД ієрархічної архітектури число ідентифікованих елементів у загальному випадку описується формулою:

$$m = n \cdot \hat{E}[\log_2 P] = \hat{E}[\log_2 N], \qquad (16.1)$$

де : P - основа (модуль відповідної позиційної системи числення*);* $\hat{E}[\cdot]$ - цілочисельна функція з округленням до більшого цілого; N – число ідентифікованих елементів баз даних; n - число розрядів коду у системі числення з модулем P; m – число двійкових розрядів ідентифікатора i–того елементу БД з діапазону $1 \le i \le N$. Графік значень N в залежності від n та P.

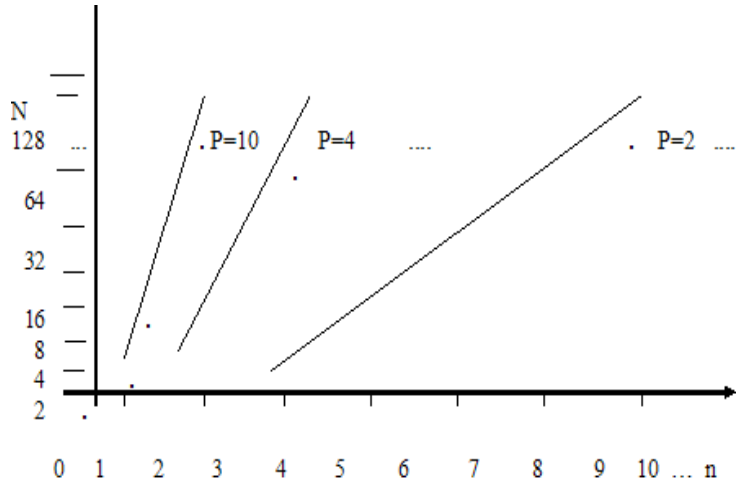


Рис.16.4. Графік числа елементів БД в залежності від параметрів n та P позиційних систем числення.

На рис.16.5-16.6 показані архітектури ієрархічних БД при різних модулях розширення.

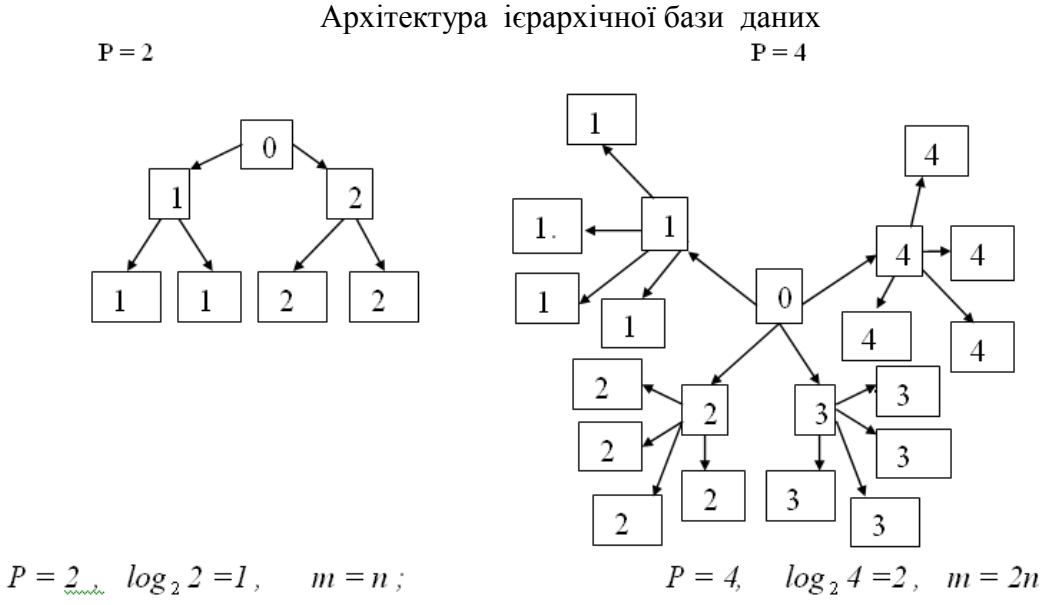


Рис.16.5. Архітектура ієрархічної БД при модулях розширення P =2 , P =4 .

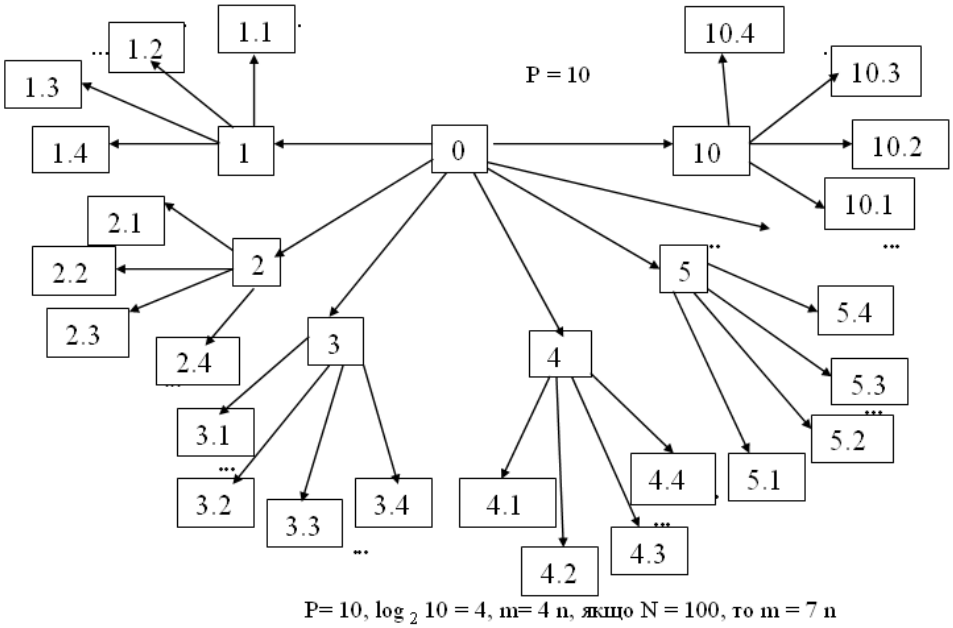


Рис. 16.6. Архітектура ієрархічної БД при модулі розширення P =10 .

Аналіз кодування даних БД в різних традиційних системах числення та застосування базису Радемахера ефективне при бінарному кодуванні

(P=2). Для інших систем числення коли $P \neq 2^k$, к = 0,1,2, ... , кодування даних БД в базисі Радемахера є надлишковим.

При використанні базису Крестенсона реалізується оптимальне кодування ієрархічної БД автономно на кожному рівні в системі взаємнопростих модулів:

$$P_1, P_2, P_3, \ldots P_i, \ldots, P_k, \ldots, \text{тобто } P_i \neq P_j .$$

Наприклад в системі базису Крестенсона виберемо модулі:

$$P_1 = 3,\ P_2 = 5,\ P_3 = 7,\ P_4 = 8,$$

що відповідають ієрархічній архітектурі БД представлений на рис.16.7.

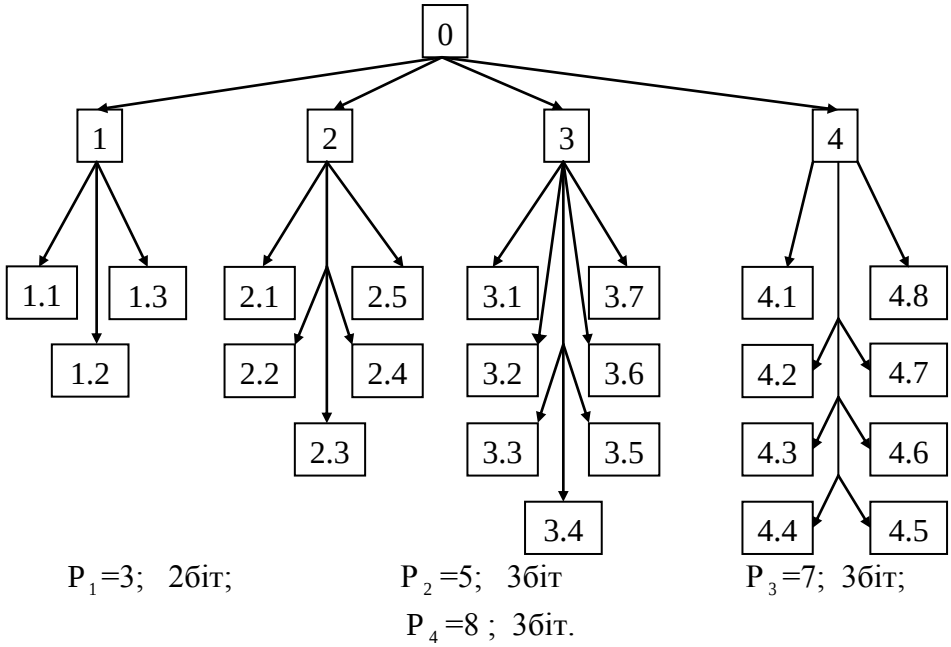


Рис.16.7. Ієрархічна архітектура БД в базисі Крестенсона.

При застосуванні СЗК досягається оптимальне кодування елементів БД у вигляді коду N_k ,який описується формулою:

$$N_k = \text{res} \sum_{i=1}^{k} b_i B_i \ (\text{mod } P) \qquad (16.2)$$

де: *res* – символ операції обчислення найменшого невід'ємного залишку ;

b_i - код елементу БД ($0 \le b_i \le P_{i-1}$);

B_i - базисні числа СЗК ; $P = \prod_{i=1}^{k} P_i$ - узагальнений модуль СЗК ;

При цьому код елементів БД в СЗК для заданої оптимальної системи модулів P_i записується у вигляді послідовності залишків.

$$N_k = (b_1, b_2, b_3, ..., b_i, ..., b_k).$$

Кодування елементів БД в базисі Крестенсона за рахунок оптимального вибору системи модулів P_i забезпечує більш компактний запис кодів.

$$0 \le N_k \le P\text{-}1,$$

які мають розрядність

$$m = \hat{E}[\log_2(P-1)] \qquad (16.3)$$

Визначаємо $P = 3 \cdot 5 \cdot 7 \cdot 8 = 1120$ звідки:

$$m = \hat{E}\,[\log_2 (1120\text{-}1)] = 11 \text{ біт.}$$

В даному випадку кодування елементів в базисі Крестенсона дає оцінку 11 біт , що на 12% менше порівняно з базисом Радемахера .

Базис Крестенсона характеризується можливостями захисту кодів залишкових класів від помилок шляхом розширення системи модулів Pi додатковим модулем P_0 по якому $b_0 = 0$, що забезпечує виявлення помилок в кодових елементів БД. Розширення системи модулів двома модулями P_k , P_0 , P_{k+1} , з відповідними залишками $b_0 = 0$, b_{k+1} , дозволяє реалізувати алгоритми виправлення помилок в кодах СЗК.

Особливістю викладеного методу кодування елементів БД є нелінійність кодів N_k які без декодування не дозволяють визначити значення b_i. В той же час дана вказана властивість може бути ефективно використана для шифрування елементів БД шляхом вибору невідомого набору модулів P_i.

При застосуванні базису Галуа реалізується рекурентний метод кодування елементів ієрархічних БД, що представлено архітектурою на рис.16.8.

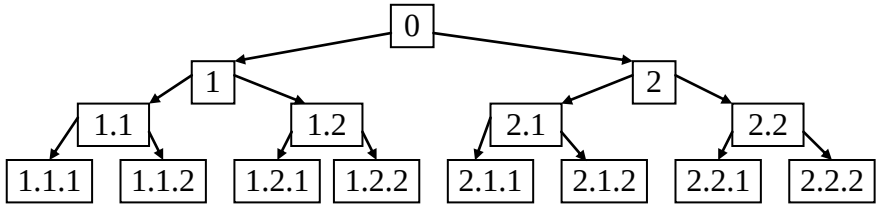


Рис.16.8. Ієрархічна БД в базисі Галуа.

На основі вертикально-рекурентної моделі БД побудованої на основі вектора базиса Галуа реалізується багаторівнева вертикально-рекурентна модель БД в базисі Галуа.

Прикладом такої реалізації є логічна структура **у**нітарного дерева СЕС у базисі Галуа (МТН корпорації „ Pallada Travell ’’USA).

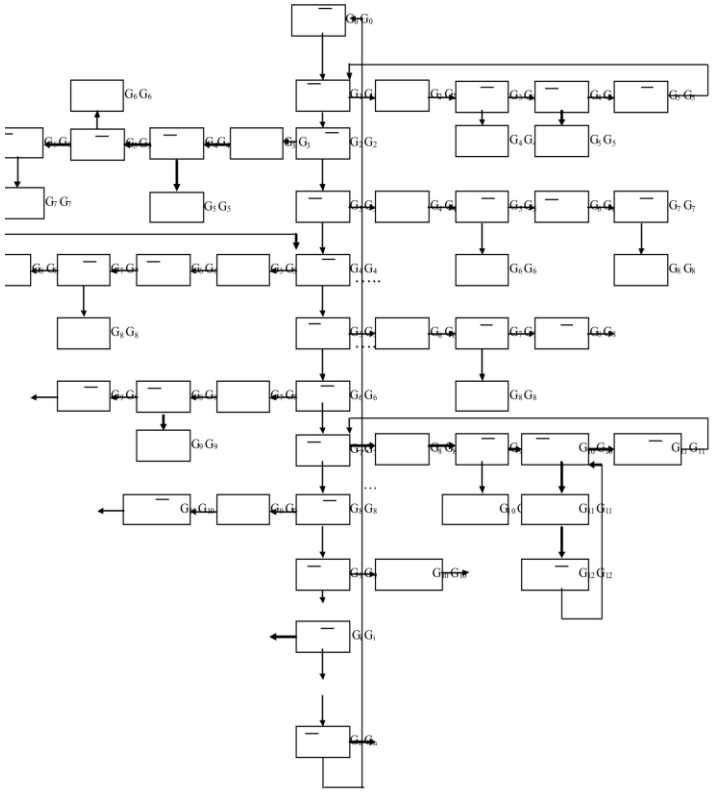


Рис.16.9. Ациклічний граф логічної структури циклів багаторівневої вертикально-рекурентної моделі БД базисі Галуа на базі унітарного дерева базису Галуа з відвітвленням руху даних СЕС корпорації „ Pallada Travell ’’ USA.

16.2.1. Критерії ефективності БД у базисі Галуа.

Важливими характеристиками ефективності БД є:

емерджентність $$K_e = \frac{N_3}{N_0};$$ (16.4)

надлишковість $$K_c = \frac{mR}{mG};$$ (16.5)

швидкість пошуку даних $V = \frac{\Delta N_i}{T}$; (16.6)

де: N_3- число зв'язків, N_0- число елементів, *mR* - ентропія ієрархічних даних в базисі Радемахера, *mG* – ентропія рекурентних даних в базисі Галуа, ΔN_i - число тактів ідентифікації даних, *T* - час доступу до даних.

Основними характеристиками оптимізації і підвищення ефективності БД є :

-здешевлення системи СУБД ;

-збільшення швидкості пошуку доступу до даних;

-зменшення або стиснення об'єму інформації ;

-розробка різноманітних типів третинних пристроїв зберігання (tertiary storage devises) даних;

-тенденції росту системи ;

-підвищення продуктивності системи ;

-паралельні обчислення;

-розвиток системи клієнт/сервер і багаторівневої архітектури;

-підвищення надійності захисту від помилок;

-захист від несанкціонованог доступу до БД тощо.

16.2.2. Аналіз ефективності кодування даних БД в базисі Галуа.

Для розрахунку ефективності фізичної структури задамо початкові умови. Нехай число елементів унітарного дерева дорівнює *m=1*, число схематично показано у вигляді горизонтальних відгалужень рівних *m=2*, число вертикальних відгалужень рівне *m=3*. Відповідно позначимо m_i та m_j відгалужень. Розрахунок об'єму даних відомої моделі унітарного дерева виконано на основі формули Хартлі

$$V_i = \hat{E}[\log_2 m_i]; \; i \in 1, n \qquad (16.7)$$

$$\sum_{i=1}^{n} V_i = \sum_{j=1}^{k} \hat{E}[\log_2 m_j] \qquad (16.8)$$

де V_i - елемент коду, $\sum_{i=1}^{n} V_i$ - загальний об'єм кодів.

Наприклад, об'єм ідентифікаційних даних нижнього абонента складає 28 біт при загальному числі 38100 абонентів.

Виходячи з максимального числа елементів БД визначимо розрядність коду ідентифікатора елементів в базисі Галуа. Якщо в БД число елементів тісно пов'язані між собою і не перевищує *n=100*, то ефективність

кодування БД в базисі Галуа можна підвищити наступним чином , шляхом кодування числа елементів в групі після коду ідентифікатора.

Звідки визначається коефіцієнт стиснення об'єму даних вертикально-рекурентної моделі БД при переході в базис Галуа по відношенні до базису Радемахера .

Наприклад: m_1=1000; m_2=2000;m_3=500;m_4=200; тоді v_1=10v_2=11; v_3=9; v_4=8 біт;

$\sum_{i=1}^{n} V_i$ = 28 000 біт , K_e = 28,000· 3,8100= 106,68 / 7,4= 14,42.

При порівнянні ефективності кодування елементів БД в різних базисах маємо ієрархічну БД з бінарним кодуванням елементів на різних рівнях*:*

N = 5000 , n = 4. Необхідно виконати адресацію 64 елементів згрупованих в три групи : 50, 10, 4.

Виходячи з максимального числа елементів БД визначимо розрядність коду індифікатора елементів в базисі Галуа

m= $\hat{E}$ [$\log_2$ 5000] = 13 біт.

Таким чином об’єм кодової інформації для ідентифікації всіх груп елементів Д описується наступним чином:

mG = 3·13+50+ 10 +4 =103 біт .

Аналогічно кодова розрядність ідентифікатора в базисі Радемахера визначається наступним чином :

mR = 13·64= 832 біт ,

звідки коефіцієнт стиснення ідентифікатора елементів БД в базисі Галуа

$$K_c = \frac{mR}{mG} = \frac{832}{103} = 8{,}077 .$$

Якщо в БД елементи тісно пов’язані між собою і не перевищує n = 100 , то ефективність кодування БД в базисі Галуа можна підвищити наступним чином , шляхом кодування числа елементів в групі після коду ідентифікатора, тобто:

mGn =3 (13+ $\hat{E}$ [$\log_2$ 100]) =3 (13+7) = 60 біт .

Звідки коефіцієнт стиснення об’єму даних при переході в базис Галуа по відношенні до базису Радемахера дорівнює:

$$Kc = \frac{mR}{mRn} = \frac{832}{60} = 13{,}86 .$$

16.3. Реляційні бази даних на основі двовимірних кодів поля Галуа.

Реляційна модель БД в даний час отримала широке застосування і ефективно використовується при створенні потужних баз даних і баз знань. Перевагою реляційних БД по відношенню до ієрархічних є забезпечення можливості формалізації перетворення даних на основі алгоритмів реляційної алгебри представлених в табл.16.1.

Таблиця.16.1

Формалізація перетворення даних

Об'єднання	$R(L)=R_1(L)\cup R_2(L)=\{r \mid r\in R_1 \vee r\in R_2\}$,	Обмеження	$S=R[L\,\theta\,M]=\{r \mid r\in R \& r[L]\,\theta\, r[M]\}$,
Перетин	$R(L)=R_1(L \cap R_2(L)=\{r \mid r\in R_1 \& r\in R_2\}$,	Декартів добуток	$Q = R\times S = \{(r, s) \mid r \in R \& s\in S;$
Різниця	$R(L)=R_1(L) - R_2(L)=\{r \mid r\in R_1 \& r\notin R_2\}$,	З'єднання	$Q=R[L\,\theta\,M]S=\{(r.s) \mid r\in R \& s\in S \& r[M]\,\theta\, s[N]\}$,
Проекція	$S=R[A_{s1},...,A_{sn}]=\{r[A_{s1},...,A_{sn}] \mid r\in R\}$,	Ділення	$R[N\div K]S=R[M]$ $(R[M]\times S[K])-)R[M]$,

Даний клас БД не в повній мірі забезпечує компактність кодування ідентифікаторів і адресацію даних. Кодування реляційних даних згідно структури відношення реляційної БД поданої на рис.1 виконується в двійковій системі числення в базисі Радемахера.

У загальному випадку для зберігання даних кортежу реляційної БД, виходячи з умови байт-орієнтованого кодування, загальний об'єм даних кортежа обчислюється згідно формули:

$$I_j = \sum_{i=1}^{n} d_{ij}$$

де d_{ij} - об'єм даних ij–го атрибута кортежу, (1б, 1Кб, 1Мб, 1Тб.,1Пб,...) ;

n = 1,2,…, - число атрибутів.

Об'єм даних відношення реляційної БД складає:

$$I_v = m\, I_j ,$$

де m =1.2…., - число кортежів.

Кількість доменів відношення реляційної БД складає:

$$D = n\ m.$$

Кожен домен реляційної БД має свій ідентифікаційний код , який будемо мати наступні параметри:

$$C_{ij} = \hat{E}[\log_2 n] + \hat{E}[\log_2 m];$$

де, $\hat{E}[\cdot]$ - цілочисельна функція з округленням до більшого цілого числа.

Таким чином повний об'єм матриці ідентифікаційних даних складає:

$$V_c = C_{ij}\ n\ m.$$

Отже оцінка співвідношення об'єму матриці ідентифікаційних даних , до об'єму даних відношення реляційної БД складає:

$$K = \frac{Vc}{Iv} \cdot 100\%.$$

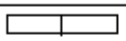
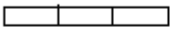
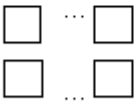
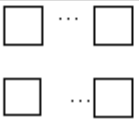
Проведемо аналіз співвідношення об'ємів матриць ідентифікаційних кодів та атрибутів реляційної БД:

$$M = \hat{E}[\log_2 V_c]2^4.$$

Результати аналізу об'ємів матриці ідентифікаторів атрибутів і кортежів відношення реляційних БД показані в табл.16.2.

Таблиця.16.2.

Результати аналізу об'ємів матриці ідентифікаторів атрибутів і кортежів відношення РБД.

i	j	d_{ij}	Архітектура РБД	c_{ij}	v_c	K	M
1	1	1		0	0	0	0
2	1	2		1	2	0,78	16
3	1	3		2	6	2,34	48
2	2	4		3	16	3,13	64
4	4	16		4	64	25	96
12	50	600		50	600	2314	201

На основі поданих аналітичних виразів та (табл.16.2) будуємо графік функції коефіцієнта *K* ефективності кодування між параметрами відношення реляційної БД в базисі Радемахера (Рис.16.10).

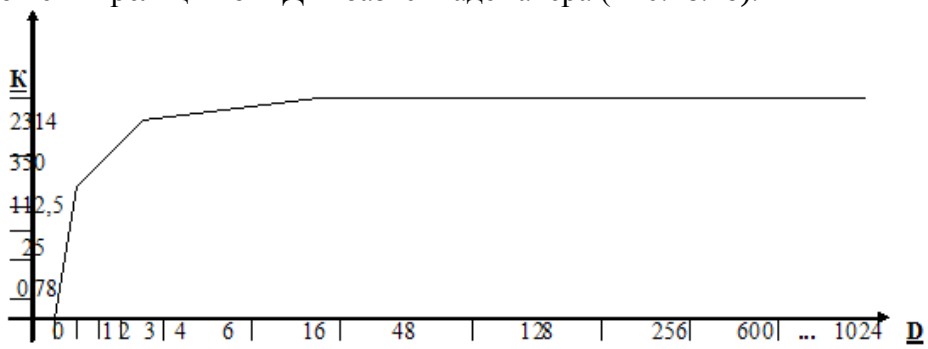


Рис.16.10. Графік залежності між параметрами відношення реляційних БД та об'ємом матриці ідентифікаційних кодів атрибутів відношення.

Визначення коефіцієнта надлишковості реляційної бази даних в базисі Радемахера при багатьох атрибутів і кортежі реляційної БД.

Коефіцієнт надлишковості реляційної БД можна обчислити згідно виразу:

$$K_1 = \frac{Cij}{Iv}\ 100\% \qquad (16.8)$$

Результати дослідження коефіцієнтів надлишковості реляційної БД показані в табл.16.3.

Таблиця 16.3.

Результати дослідження коефіцієнтів надлишковості РБД.

i	j	n	m	K_1 %
1	1	1	1	0
1	2	1	2	50
1	3	1	3	66
1	4	1	4	50
1	8	1	8	36
1	128	1	128	30
1	256	1	256	37
1	2	1	2	50
2	2	2	2	44
2	3	2	3	83
4	4	4	4	50
8	16	8	16	30
12	50	12	50	13

З допомогою досліджених результатів М та визначених значень коефіцієнта К побудуємо графік функції коефіцієнта ефективності кодування БД на основі кодів базису Радемахера.

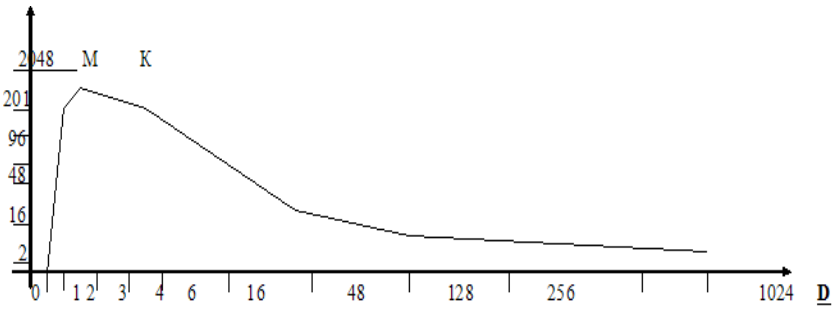


Рис.16.11. Графік залежності функції коефіцієнта ефективності кодування БД в базисі Галуа при великих масивів даних.

Базис Радемахера породжує двійкову систему числення. В базисі Радемахера працюють всі сучасні комп'ютерні системи.

На основі базисів Радемахера застосовується одномірне кодування:

(D XXX. XXX) R D 10.10 i j (16.9)

Для кодування чисел в базисі Радемахера в діапазоні *0-N* необхідно $log_2 N$ розрядів двійкових чисел.

При кодуванні базису Галуа на кожен N_i число припадає 1 біт Галуа А для кодування чисел *N* з вектору необхідно рекурентно вибрати m-біт даних що засвідчує максимальну упаковку інформації в базисі Галуа.

Для одномірного кодування застосовується формула генерування одномірного коду поля Галуа:

$$G_{i+1} = G_i \bar{\oplus} G_{i-n}$$

де G_i - текучий біт Галуа, $\bar{\oplus}$ -додавання по mod2, G_{i-n} -зміщення розрядів біт Галуа.

На основі базису Галуа застосовується двомірне кодування:

(D X. X.) G D 1.

Для двомірних кодів поля Галуа аналітичний вираз генератора має вид:

$$G_{i+1,j+1} = G^{\circ}_{i+1} \wedge G^{\circ}_{Gj+1},$$

де $G^{\circ}_{i+1} = G_i \bar{\oplus} G_{i-n}$; $G^{\circ}_{j+1} = G_j \bar{\oplus} G_{j-m}$.

На основі порівняння базису Радемахера і базису Галуа побудуємо відношення реляційних БД двомірних кодів в базисі Галуа. табл.16.4.

Таблиця16.4.

Відношення реляційних баз даних в базисі Галуа.

11	11	11	11	01	11	01	11	01	01	11	01	01	01	01
11	11	11	11	01	11	01	11	01	00	11	01	01	01	01
11	11	11	11	01	11	01	11	01	00	11	01	01	01	01
11	11	11	11	01	11`	01	11	01	00	11	01	01	01	01
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00
11	11	11	11	01	11	01	11	01	01	11	01	01	01	01
10	10	10	10	00	10	01	10	00	00	10	00	00	00	00
11	11	11	11	01	11	01	11	01	01	11	01	01	01	01
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00
10	10	10	10	00	01	00	10	00	00	10	00	00	00	00
11	11	11	11	01	11	01	11	01	01	11	01	01	01	01
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00
10	10	10	10	00	10	00	10	00	00	10	00	00	00	00

З отриманих результатів дослідження можна зробити порівняльну характеристику одномірного коду ідентифікатора атрибутів даних в базисі Радемахера з двомірним кодом ідентифікатора атрибутів даних в базисі Галуа в табл.16.5 для i =1,2, ..., n; j = 1,2, …, m; подані двомірний код Галуа.

Таблиця 16.5.

Порівняльна характеристика одномірного коду Радемахера з двомірним кодом Галуа.

Одномірний код ідентифікатора атрибутів даних в базисі Радемахера		Двомірний код ідентифікатора атрибутів даних в базисі Галуа
i	j	i j
0 0 0	0 0 0	0 0
0 0 1	0 0 1	0 0
. . .	. . .	. . .
1 1 1	1 1 1	1 1

Оцінка аналізу ефективності кодування ідентифікаторів РБД двомірними кодами Галуа, виконуємо на основі виразу:

$$K_e = \log_2 \left[\frac{E \cdot [\log_2 n] + E \cdot [\log_2 m]}{2} \right], \qquad (16.10)$$

де. n, m – число атрибутів та кортежів реляційної БД.

На рис.16.15 показані графіки ефективності кодування ідентифікаторів реляційної БД на основі рекурентних двомірних послідовностей Галуа.

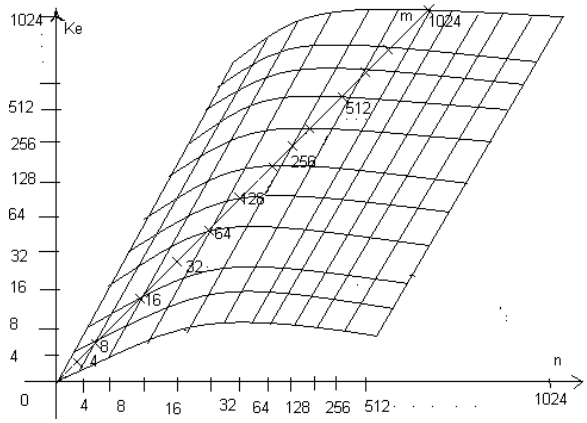


Рис.16.15. Графіки ефективності кодування ідентифікаторів реляційної БД на основі рекурентних двомірних послідовностей Галуа.

Дослідження теоретико–числових базисів та їх застосування при організації БД, показують, що традиційний спосіб ідентифікації та кодування елементів БД в базисі Радемахера є неефективним, надлишковим і завадонезахищеним, крім того традиційний метод є незахищеним від несанкціонованого доступу, що достатньо ефективно реалізується в базисі Галуа.

РОЗДІЛ 17

АЛГОРИТМИ ПЕРЕТВОРЕННЯ ДАНИХ НА ОСНОВІ КОДІВ ПОЛЯ ГАЛУА

17.1. Алгоритми стиснення та декодування даних представлених кодами поля Галуа.

Важливим етапом написання програмного забезпечення методу стиснення даних, представлених гармонічними сигналами, на основі кодів Галуа є розробка алгоритму. Необхідно визначити структуру системи (модулі), 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, можливому в процесі розв'язку задачі.

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

- модуль архівування (рис.17.1);
- модуль розархівування (рис.17.2);

Модуль архівування 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 (рис.17.2) 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.

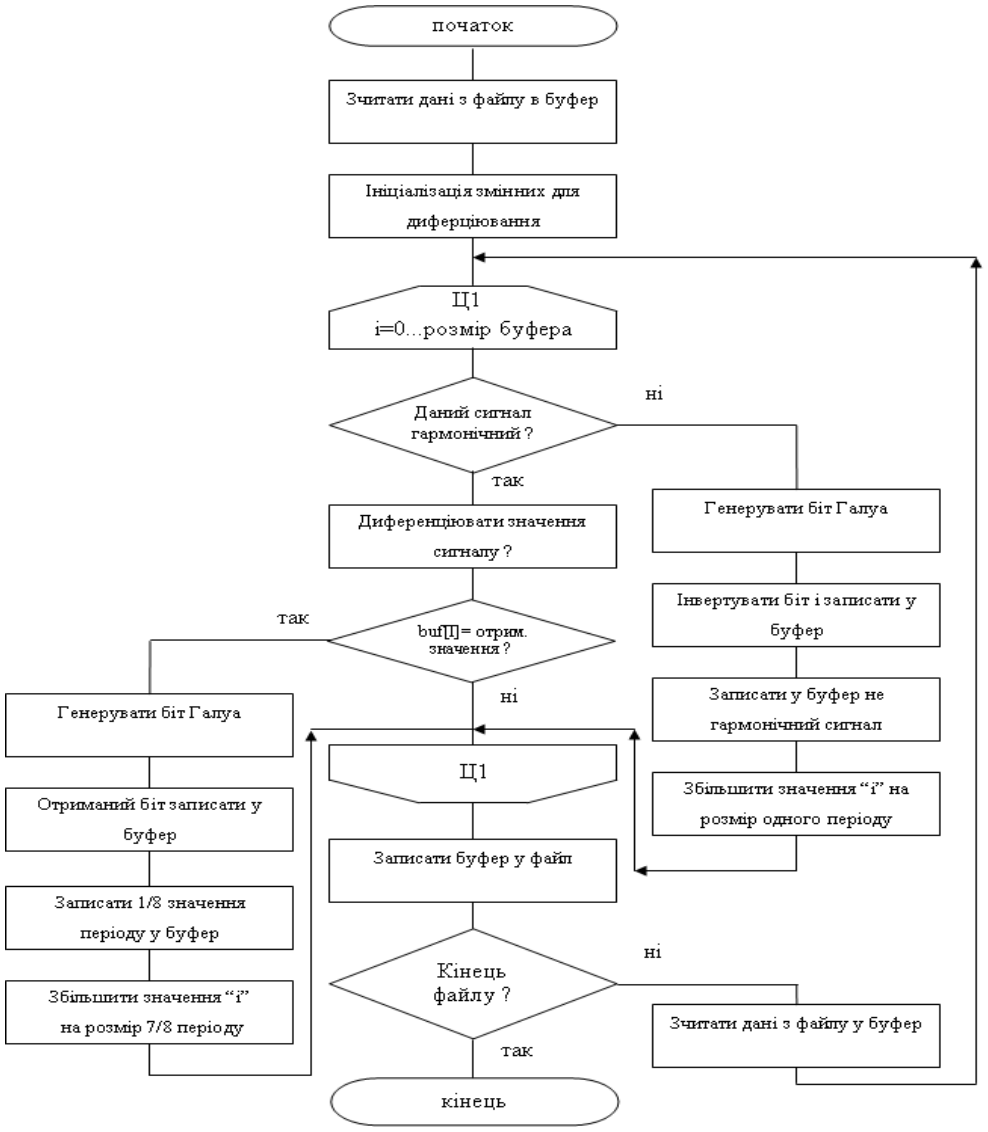


Рис.17.1. Схема алгоритму архіватора.

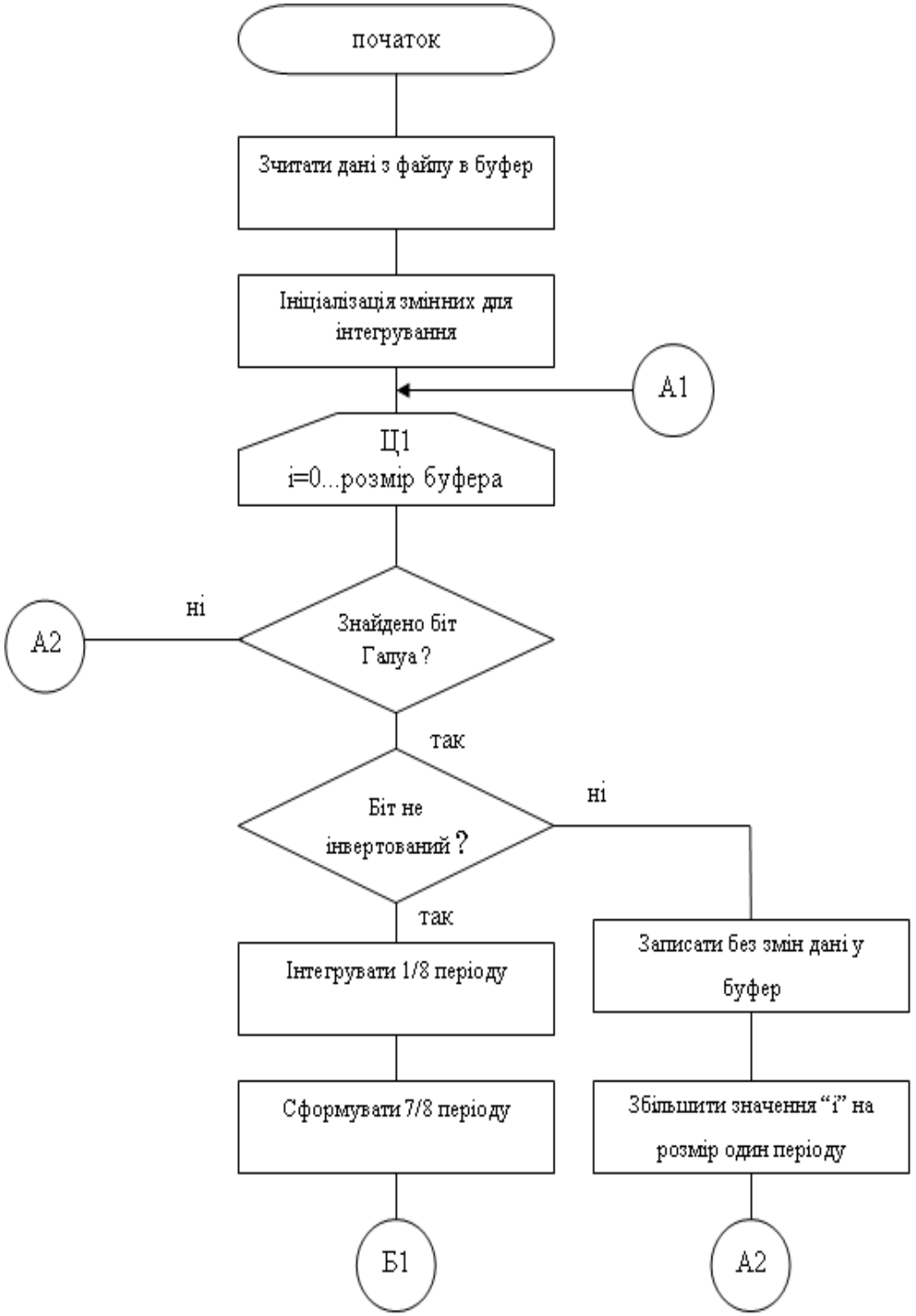


Рис.17.2. Схема алгоритму розархіватора.

Продовження рис. 17.2.

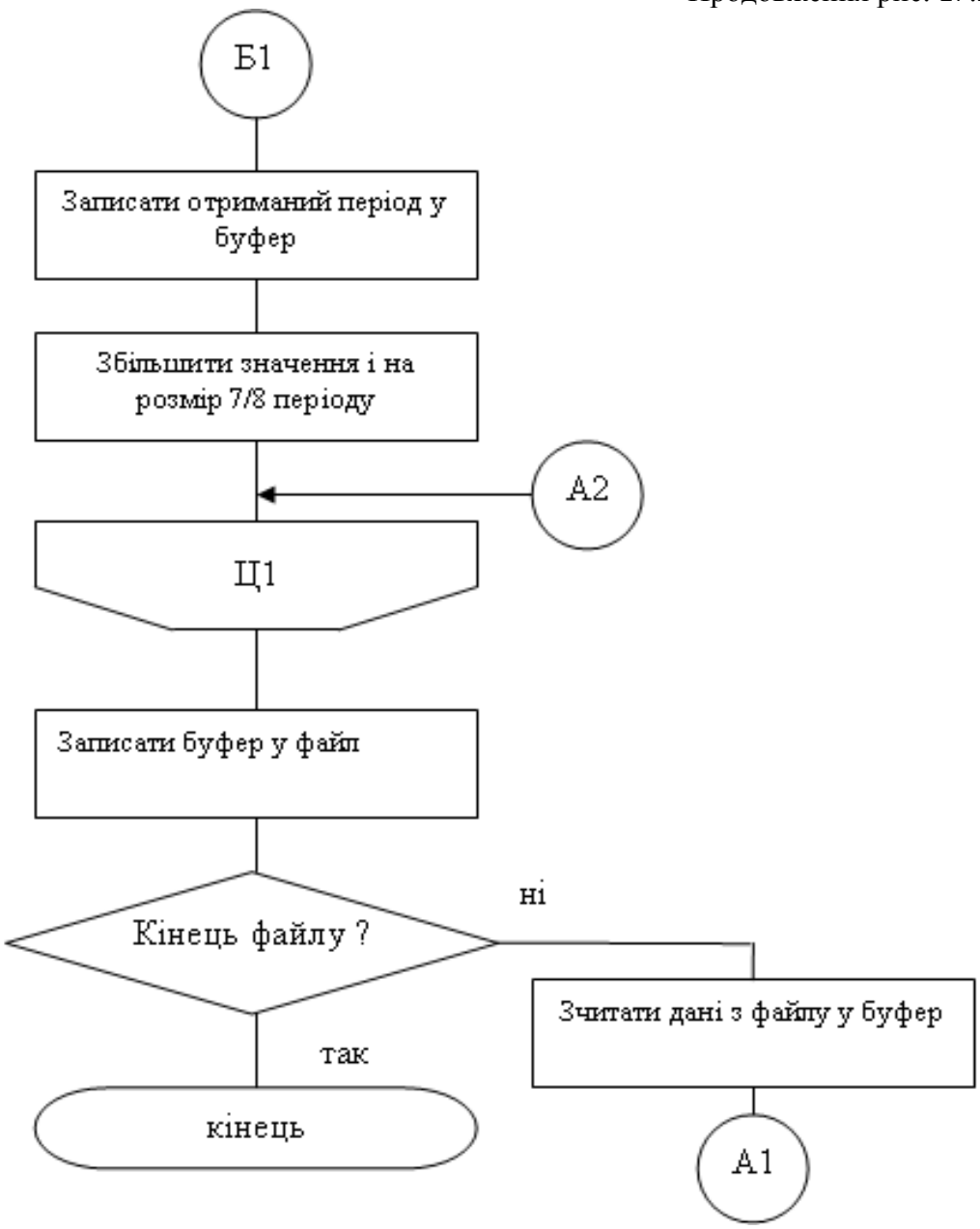


Рис.17.2. Схема алгоритму розархіватора.

17.2. Алгоритми роботи цифрових приймачів сигнальних коректуючих кодів поля Галуа.

На рис.17.3 показана блок-схема програмного модуля циклу перевірки наявності *n*-бітової послідовності стартових сигналів.

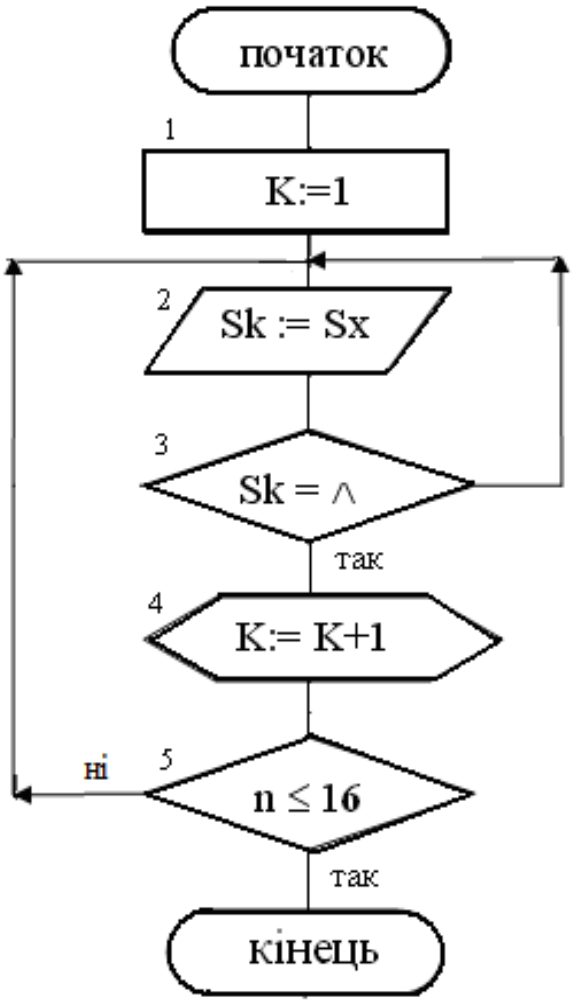


Рис.17.3. Алгоритм роботи програмного модуля стартових сигналів «start».

Даний алгоритм включає в себе наступні блоки:

1. Початкові параметри циклу перевірки n-бітової послідовності start сигналів.

2. Ввід сигналу Sx для аналізу.

3.Перевірка наявності в потоці даних сигналу, який не відповідає стартовим сигналам.

4-5.Організація циклу.

Аналогічно реалізується програмний модуль перевірки наявності сигналів блокової синхронізації «stop», який може бути присутній в окремих випадках або може бути опущений в інформаційних каналах з низьким рівнем завад, що дозволяє зменшити надлишковість об'єму даних. Ефективність зменшення надлишковості таких інформаційних потоків є найбільш ефективною на низових рівнях комп'ютерних систем, які реалізуються пакетами невеликої довжини. В алгоритмі роботи блоку «stop» даний модуль відрізняється від блоку «start» оператором 3, який виконує перевірку Sk = "$\vee$".

Алгоритм оброблення сигналів, які формуються інформаційним потоком у вигляді ПСК, описується роботою наступних програмних модулів, згідно блок-схеми (рис.17.4):

1. Перевірка послідовності сигналів «start» , яка представлена на рис.17.3.

2. В даному програмному модулі вводяться стартові характеристики даних D; регістра стартової позиції генератора Галуа G_0 та лічильника помилок $*_i = 0$.

3. Описує алгоритм генерування коду поля Галуа, який використовується для перевірки правильності інформаційного потоку даних, що представляється сигналами Sx.

4. Ввід поточного сигналу Sx.

5-6, 10. Перевірка правильності приймання та формування бітів даних одиниці.

7-8, 11. Перевірка правильності приймання бітів даних нулів.

9. Реєстрація числа помилок $*_i$.

12. Реєстрація та формування пакету даних.

13-14. Цикл перевірки об'єму даних.

15. Перевірка послідовності сигналів «stop».

16. Перевірка на наявність помилки

17-18. Вивід даних та вивід помилок.

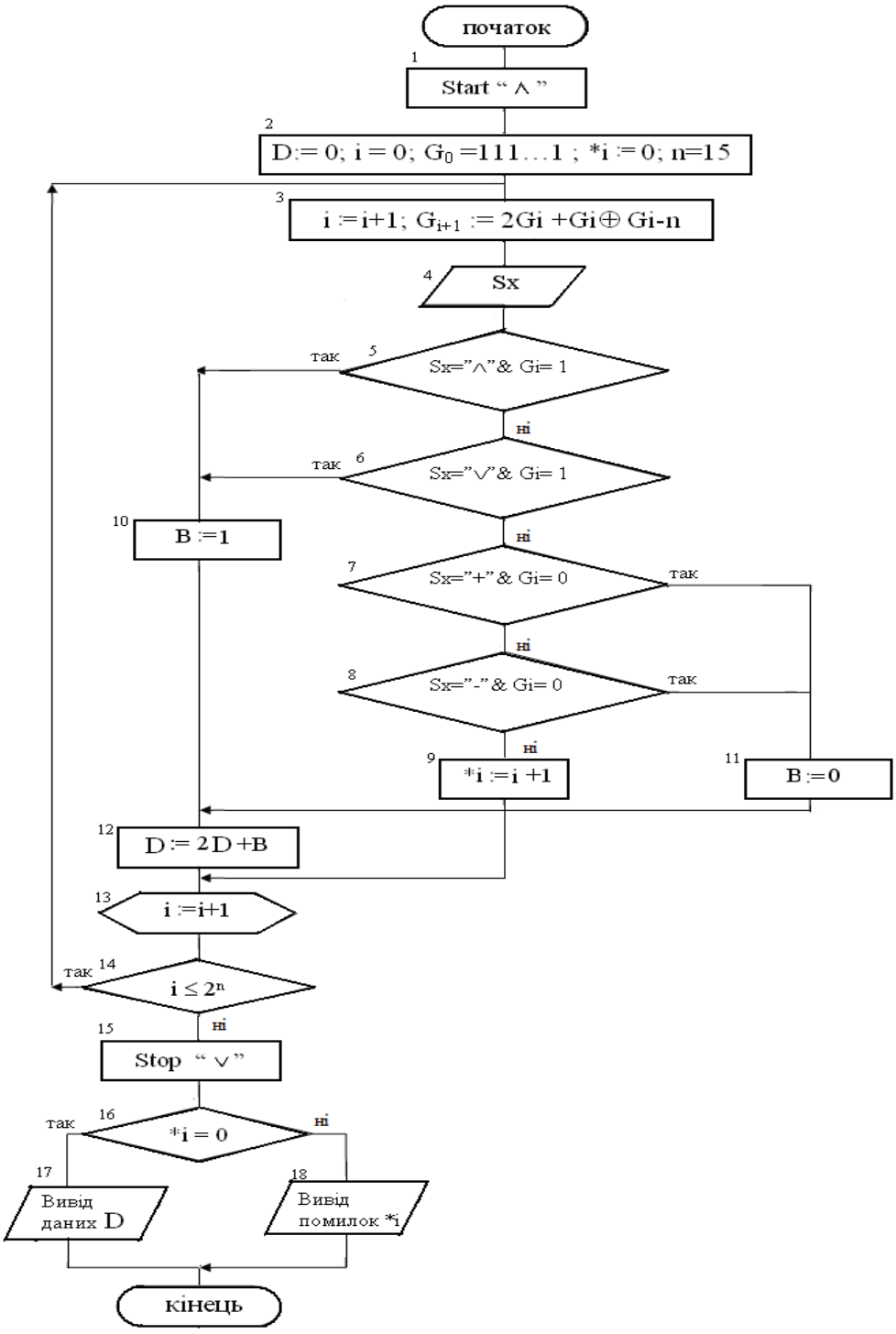


Рис.17.4. Алгоритм роботи програмних модулів ПСК.

Алгоритм оброблення сигналів, які формуються інформаційним потоком у вигляді НРСК представлено на рис.17.5:

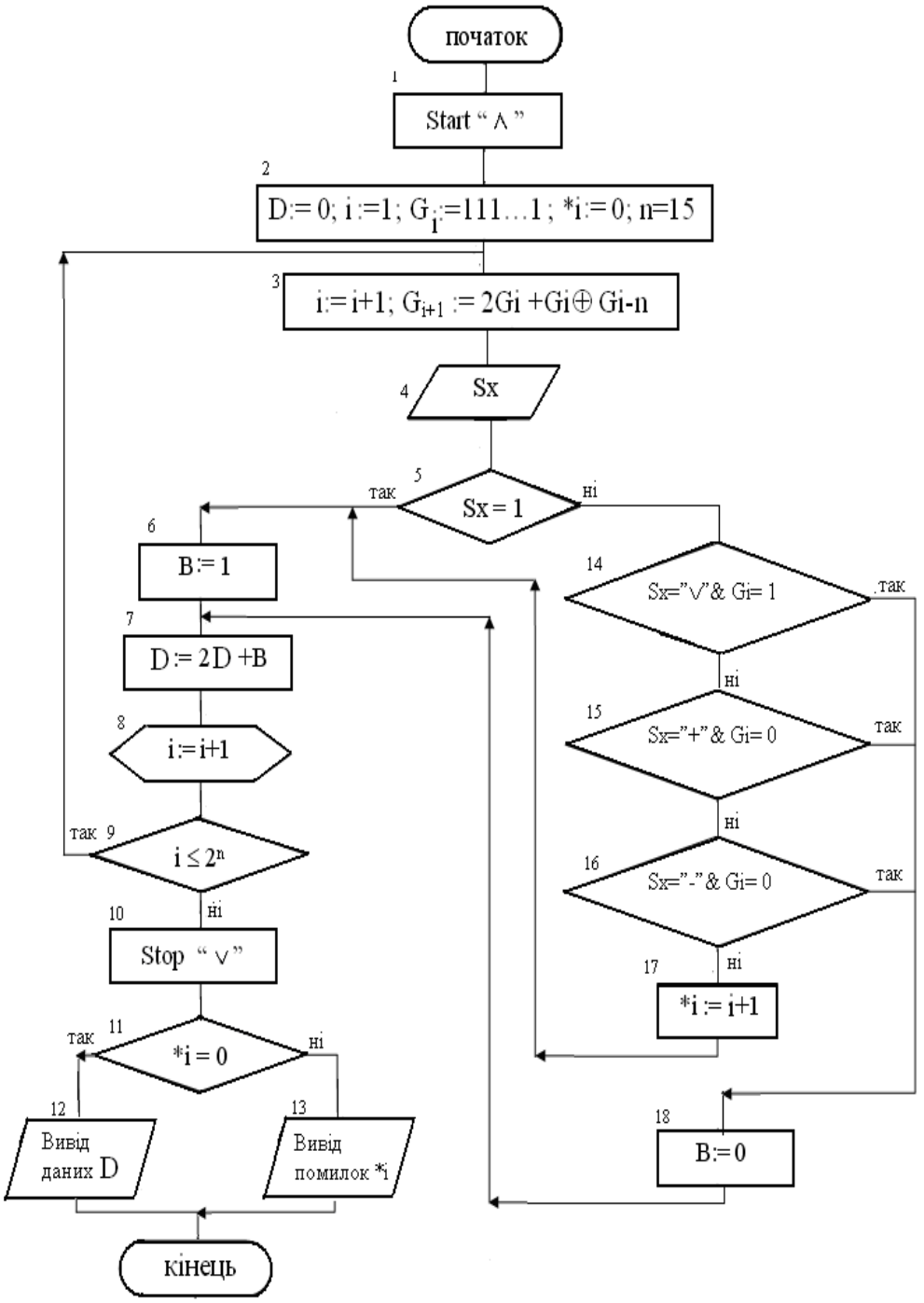


Рис.17.5. Алгоритм роботи програмних модулів НРСК.

Робота наступних програмних модулів описується наступним чином:

1-4. Аналогічні алгоритму на рис.17.3.

5-6. Перевірка наявності сигналу фронту наростання Sx = "∧", і формування біту даних одиниці.

7-13. Аналогічні алгоритму на рис.17.3.

14-16. Перевірка приймання правильних сигналів, які кодують нуль потоку даних.

17. Реєстр помилок.

18. Формування біту даних нуль.

Алгоритм оброблення сигналів, які формуються інформаційним потоком у вигляді РССК, описується наступними програмними модулями, згідно блок-схеми (рис.17.6):

1. Перевірка послідовності сигналів «start» .

2. В даному програмному модулі вводяться стартові характеристики даних D; регістра стартової позиції генератора Галуа G_0 та лічильника помилок $*_i = 0$.

3. Ввід сигналу Sx.

4. Перевірка правильності приймання та формування бітів даних одиниці по фронтах спаду або наростання.

5. Генерування коду поля Галуа, який використовується для перевірки правильності інформаційного потоку даних, що представляється сигналами Sx.

6-8. Формування біта одиниці.

9. Підрахунок відповідного числа одиниць N_1.

10. Перевірка правильності приймання та формування бітів даних нуля по верхніх та нижніх потенціалах.

11. Генерування коду поля Галуа, який використовується для перевірки правильності інформаційного потоку даних, що представляється сигналами Sx.

12-14. Формування біта нуля.

15. Підрахунок відповідного числа нулів $N_{0.}$

16. Формування та реєстрація потоку даних.

17-18. Цикл перевірки об'єму даних.

19. Перевірка послідовності сигналів «stop».

20. Перевірка комплектності прийнятого пакету даних.

21. Перевірка на наявність помилки в сигналах «stop».

22-23. Вивід даних та вивід помилок.

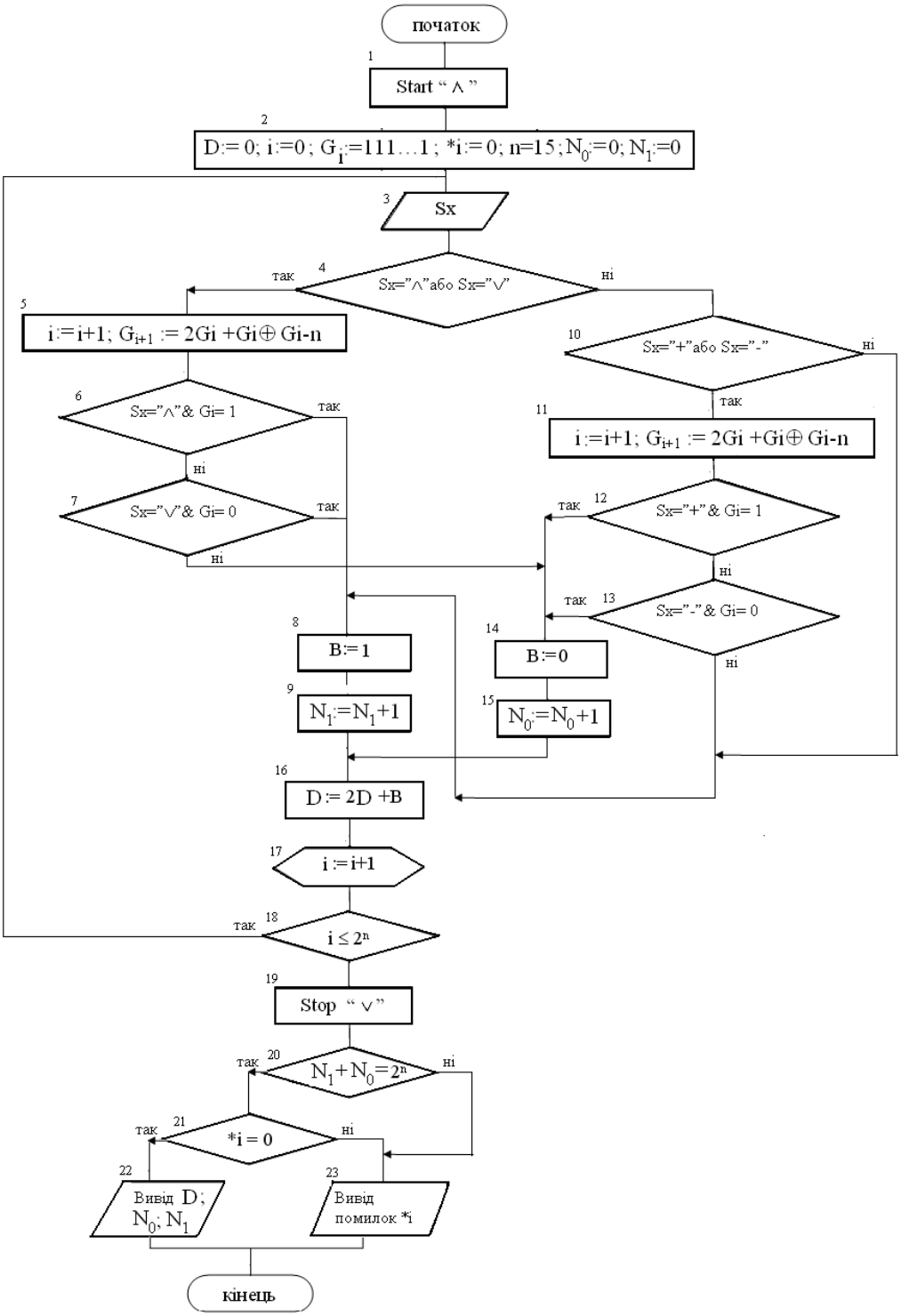


Рис.17.6. Алгоритм роботи програмних модулів РССК.

Алгоритм КССК, в якому біти даних одиниць керуються фронтами наростання чи спаду (“∧” або “∨”), тобто забезпечують високоякісну бітову синхронізацію, використовує сигнал синхронізації “S”, при формуванні сигнальних даних, тільки для виключення повторних символів “+” та “-“, які в свою чергу знижують рівень бітової синхронізації. Тому при формуванні КССК з симетричним використанням сигналів синхронізації S на основі використання модемів з фазовою частотою чи іншою маніпуляцією, після кожного прийняття біта даних в алгоритмі необхідно аналізувати факт його повторення, що представлено в блок-схемі на рис.17.7.

Алгоритм роботи програмних модулів методу КССК (рис.17.8) такий, як і в РССК, з відмінністю в тім, що відбувається додаткова перевірка на наявність повторення симмолів.

У блоці 4 відбувається перевірка наявності вхідного сигналу “S’ (синхро), при цьому в операторі 5 присвоюється значення попереднього біту даних. Сигнал “S” кодує повтор попереднього біту даних незалежно від біту Галуа.

Оператори 6,7,12,18 – виконують формування відповідних прийнятих бітів даних та підрахунок відповідного числа одиниць N_1 та числа нулів N_0.

Реалізація розроблених алгоритмів приймання різних сигнальних коректуючих кодів у вигляді пакету програм на мові с++ наведена на рис.17.7.

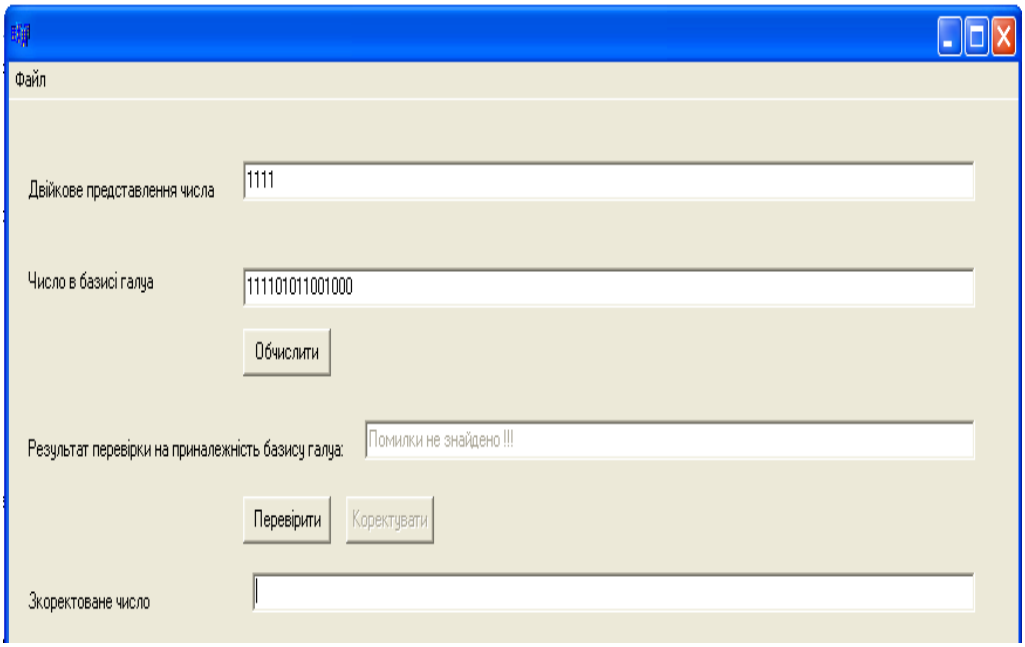


Рис. 17.7. Приклад програмного формування коду Галуа.

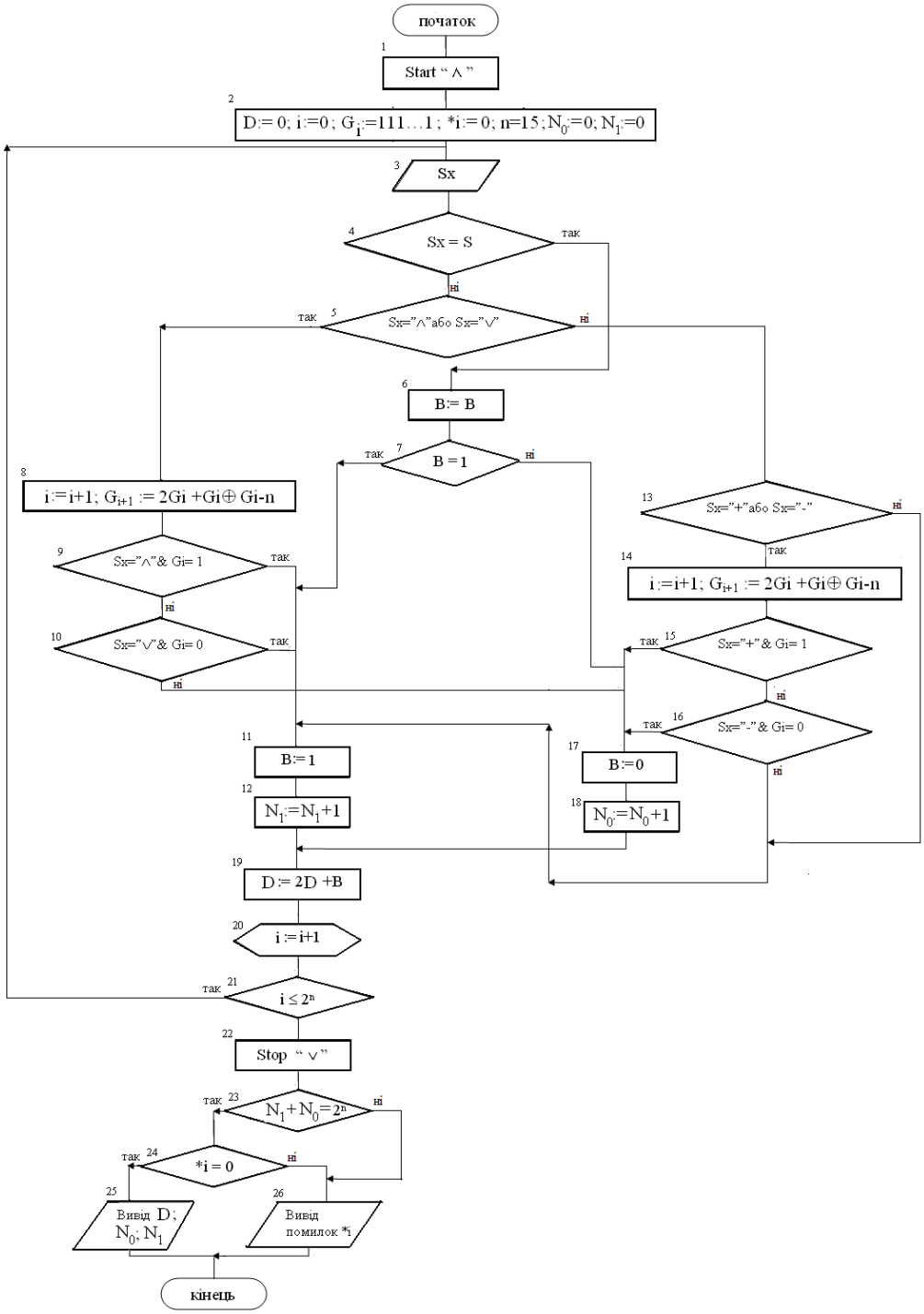


Рис.17.8. Алгоритм роботи програмних модулів КССК.

РОЗДІЛ 18

КОДИ ПОЛЯ ГАЛУА ТА ІНФОРМАЦІЙНА ТЕОРІЯ ДНК

18.1. Генерування рекурентних кодових послідовностей в різних полях Галуа.

Кодові ключі рекурентних послідовностей Галуа визначаються на основі теорії незвідних поліномів Галуа.

Базовим рівнянням генерації коду поля Галуа є рекурентна форма:

$$G_{i+1} = a_i G_i \oplus a_{i-1} G_{i-1} \oplus ... \oplus a_{i-n} G_{i-n}, \qquad (18.1)$$

де $a_i \in \overline{0,1}$ - логічні значення вектора кодового ключа; G_i - логічні значення елементів кодона КПГ $FG(\binom{n}{2})$; n - довжина кодона.

Слід зауважити, що число одиничних елементів кодового ключа завжди парне. У табл.18.1 приведені кодові ключі для різних n.

Таблиця 18.1.

Кодові ключі $FG(\binom{n}{2})$.

n	a_i
0 1 0	101; 1 10; 1001; 1 100; 10101; 1 0010; 100001; 1 10000; 110000; 1 000100; 101110 00; 100010 000; 100100 0000; 101000 00000; … 100100000000000 00000.

Для багаторівневих КПГ кодові ключі складніші. Наприклад: $n = 3; (G_2^3)$ відповідно кодовий ключ описується рівнянням

$$G_{i+1} = (2G_i + G_{i-1} + 2G_{i-2}) \bmod 3$$

і дозволяє отримати рекурентну послідовність коду поля Галуа максимальної довжини по $\bmod 3$ у вигляді:

$$22212010110021112102020001.$$

На основі такого коду Галуа можна отримати його модифікації та представлення наступними способами:

1) у зворотному порядку зчитування

$$10002202012111200110102122 2;$$

2) у центрованому вигляді шляхом заміни символів: $0 \to 2$; $1 \to 1$; $2 \to -1$:

$$---0-+0+00++-000-+-+----++0; \qquad (18.2)$$

$$0+++--+-+0-000-++00+0+-0---;$$

3) згорткою елементів рекурентної послідовності коду Галуа (18.2) довжиною $n = 3^3 - 1$ шляхом обчислення авто кореляційної функції згідно виразу у базисі ортогональних функцій Хаара (рис.18.1):

$$T_{xx}(j) = \frac{1}{n}\sum_{i=1}^{n} har\overset{\circ}{x}_i \cdot har\overset{\circ}{x}_{i+j},$$

де $har\overset{\circ}{x}_i = \begin{cases} +1 & \overset{\circ}{x}_i > 0 \\ 0 & \overset{\circ}{x}_i = 0 \\ -1 & \overset{\circ}{x}_i < 0 \end{cases}$.

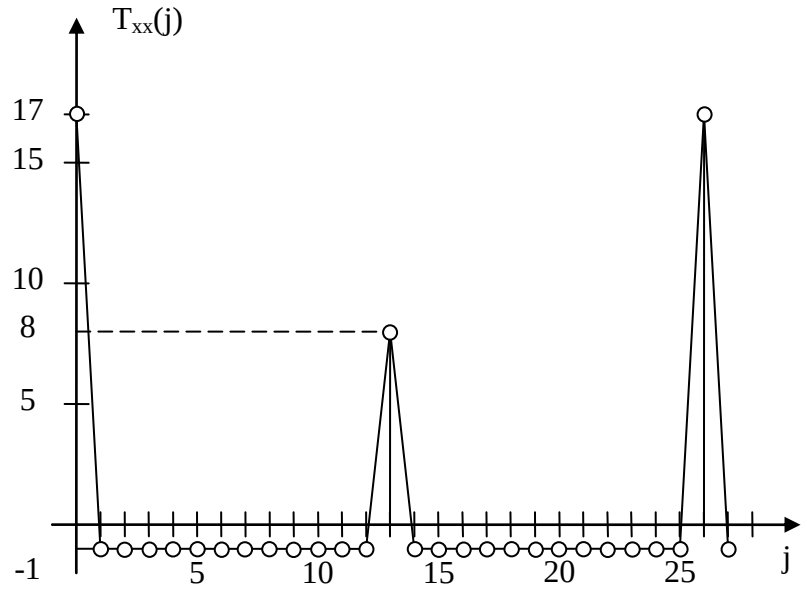


Рис.18.1. Автокореляційна функція коду Галуа $GF\begin{pmatrix}3\\3\end{pmatrix}$.

Таким чином можуть бути сформовані рекурентні послідовності кодів поля Галуа при різних модулях (m) та різних довжинах кодонів (n). Тобто:

– довжина коду поля Галуа $n = m^2 - 1$;

– число різних кілець Галуа

$$M = m^{m^{n-1}} - n. \qquad (18.3)$$

В табл.18.1 приведені розраховані дані числа кодів Галуа у полі $GF(\binom{3}{3})$, тобто $m = 2$.

Таблиця 18.1.

Число кодів Галуа у полі $GF(\binom{3}{3})$.

n	2	3	4	5	6	7	8	9	10	11
M	1	2	2	6	6	18	26	48	60	176
n	12	13	14	15	16	17	18	19	20	21
M	244	630	756	1300	2049	7710	7776	27594		

Дані табл.18.1 свідчать про швидке зростання рекурентних кодів поля Галуа при зростанні числа рівнів m, що відповідає багатозадачності відповідної групи поля Галуа, а також збільшення довжини кодонів n.

Серед досліджуваних кодів поля Галуа особливими кореляційними властивостями характеризуються так звані комплементарні коди ССК (Complementary Code Keying), які ефективно використовуються у безпровідних технологіях передавання даних, захищених від помилок методом DSSS з пропускною здатністю 11 Мбіт/с.

Кодові послідовності ССК служать аналогом кодів Баркера і характеризуються тією властивістю, що сума їх автокореляційних функцій для будь-якого циклічного зсуві відмінного від нуля завжди дорівнює «0» і максимальна при нульовому зсуву. При цьому дві кінцеві двійкові послідовності однакової довжини називають табулюючими, якщо число пар подібних елементів в одній рівне числу пар відмінних – іншій (рис.18.2).

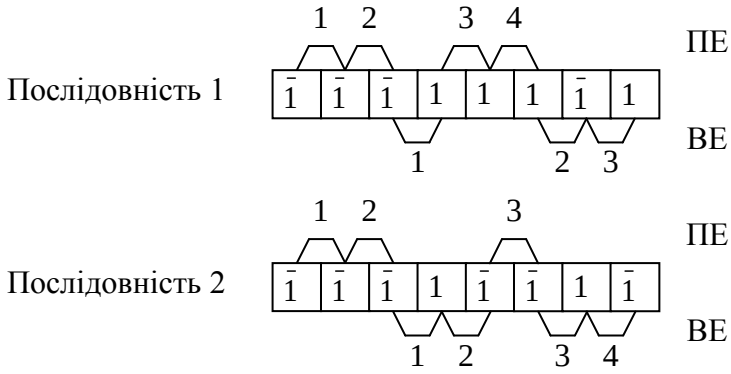


Рис.18.2. Комплементарні коди ССК методу DSSS. ПЕ – подібні елементи, ВЕ – відмінні елементи.

Приклад кореляційної згортки комплементарних кодів та визначення суми їх кореляційних функцій згідно виразів (18.4) приведений у табл.18.2.

$$C_j = \sum_{i=1}^{n-1} a_i \cdot a_{i+j}\,;\; d_j = \sum_{i=1}^{n-1} b_i \cdot b_{i+j}\,;\; j = \overline{0,n}\,;\; C_j + d_j = \begin{cases} 0 & j \neq 0 \\ 2n & j = 0 \end{cases}. \quad (18.4)$$

Таблиця 18.2.

Особливості кодування ДНК.

	Послідовність 1									Послідовність 2									d_j	$c_j + d_j$
	Код a									Код d										
0	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1	$\bar{1}$	1	8	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	8	16
	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1	$\bar{1}$	1		$\bar{1}$	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	$\bar{1}$	1	$\bar{1}$		
1	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1	$\bar{1}$	1	0	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	0	0
	1	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1	$\bar{1}$		$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	$\bar{1}$	1		
2	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1	1	1	0	1	$\bar{1}$	$\bar{1}$	1	1	$\bar{1}$	1	$\bar{1}$	0	0
	$\bar{1}$	1	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	1	1		$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	1	$\bar{1}$	$\bar{1}$		

3	$\overline{1}$	$\overline{1}$	$\overline{1}$	1	1	1	$\overline{1}$	1	-4	$\overline{1}$	$\overline{1}$	$\overline{1}$	1	$\overline{1}$	$\overline{1}$	1	$\overline{1}$	4	0
	1	$\overline{1}$	1	$\overline{1}$	$\overline{1}$	$\overline{1}$	$\overline{1}$	1		$\overline{1}$	1	$\overline{1}$	$\overline{1}$	$\overline{1}$	$\overline{1}$	1	$\overline{1}$		

18.2. Особливості кодування нуклеотидних послідовностей ДНК.

Приведені властивості комплементарних кодів визначають особливу цікавість дослідження відповідних властивостей комплементарних нуклеотидних послідовностей ДНК, яку М. Д. Франц-Каменецький називає «найбільш головною молекулою» біологічних видів.

Слово «ген», яке дало початок сучасної генетики, було введено В.Йогансеном у 1910 році і відносилося до гіпотетичної одиниці інформації, що регулює наслідування індивідуальних ознак організму. В перших дослідженнях генами оперували як абстрактними статистичними поняттями, оскільки не було ніякої інформації відносно хімічної природи ознак, які вивчалися.

Молекула дезоксирибонуклеїнової кислоти (ДНК) була відкрита швейцарським лікарем І.Ф.Мішером у 1868 році. Потім Морганом було встановлено, що ДНК міститься у хромосомах. Але шлях до її визнання в якості «головної інформаційної молекули» був ще досить довгим. Тому в 20-30-х роках XIX століття у дослідників утвердилася думка, що ДНК – це регулярний полімер, який складається зі строго повторимих четвірок мономірних ланок (А-аденінової, G-гуаніновою, Т-тіміновою, і С-цитозинової) і тому ця молекула не може нести генетичну інформацію. Вважали, що ДНК виконує в хромосомах якусь структурну роль, а гени складаються з білків, які входять у склад хромосом. Першою роботою, в якій було доведено, що речовина спадковості, або гени, є саме молекула ДНК була науковою працею Евері.

Генетики виявилися перед вибором – або не повірити законам Евері, або признати , що речовиною спадковості виявляється не білок, як усі вважали, а ДНК. Дослідам Евері було дано наступне пояснення: ДНК, звичайно ніяких генів не містить і містити не може. Але вона може викликати мутації, тобто змінювати гени, які, як їм належить складаються з білків. Теорією, яка вирішила, що ж саме спостерігав Евері у своїх дослідах була спіральна модель будови ДНК, яку відкрили англійські (нобелівські лауреати) Уостсон і Крік.

Звідси згідно моделі Уотсона і Кріка молекула ДНК складається з двох комплементарних полімерних ланцюгів (рис.18.3).

Рис. 18.3. Комплементарна структура коду ДНК.

Всередині кожного полімерного ланцюга атоми скріплені дуже потужними ковалентними зв'язками, а між комплементарними ланцюгами діють відносно слабкі взаємодії, подібні до тих, які утримують молекули одну біля одної у кристалах.

Незважаючи на відкриття структури молекули ДНК деякі генетики продовжували фанатично триматися за білки з аргументацією, що « така складна штука, як життя , не могла бути у своїй основі влаштована так просто».

На рубежі 50-х і 60-х років Френсіс Крік і його співробітники довели, що код ДНК триплетний. Тобто одній з 20-ти відомих амінокислот відповідає послідовність із трьох нуклеотидів (рис.18.4).

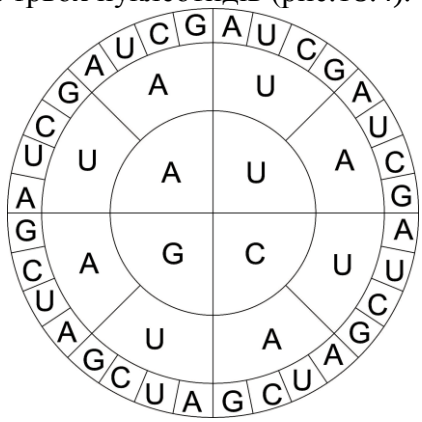


Рис. 17.4. Генетичний код ДНК.

Таким чином було експериментально доведено, що число можливих кодонів $K=4^3=64$.

До 1961 року стало ясно, що код ДНК триплетний рекурентний(тобто зчитування відбувається кодон за кодоном), і що він у якості розділових знаків містить ініціюючі (стартові) та термінуючі (стопові) кодони. Таким чином була встановлена базова структура коду генів у вигляді послідовності промотора, стартового кодона, послідовності кодонів коду білка і стопового кодона.(рис.18.5).

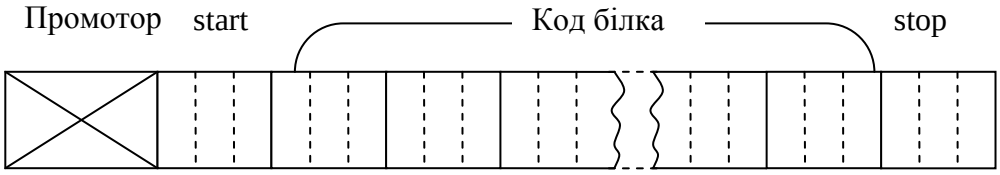


Рис. 18.5. Структура гена

Крім того досліджено, що в РНК (рибонуклеїнова кислота) грає роль копії гена у вигляді одиничного ланцюга (рис.18.6), в якому нуклеотид Т-тіміновий замінений на U-урадіновий.

Рис. 18.6. Фрагмент транспортної РНК

У 1961 році М. Нюренберг і Дж. Маттеі першими розшифрували (ідентифікували) кодон UUU, якому відповідає амінокислота феніланін.

Важливим відкриттям, які дали основу синтезу білків та генної інженерії було відкриття американцем Г. Теміном (нобелівським лауреатом) у 1970 році фермента ревертази, який синтезує ДНК на основі РНК.

Потім відкриття рекстріктази, яка пізнає різні послідовності нуклеотидів і дозволяє «розрізати» ДНК на окремі фрагменти і лігази, яка дозволяє «зшивати», тобто відновлювати строго ідентично розчленовану ДНК або утворювати нові рекомбінантні ДНК з різних фрагментів.

На рис.18.7 показана схема зчитування та генерації білка рибосомою.

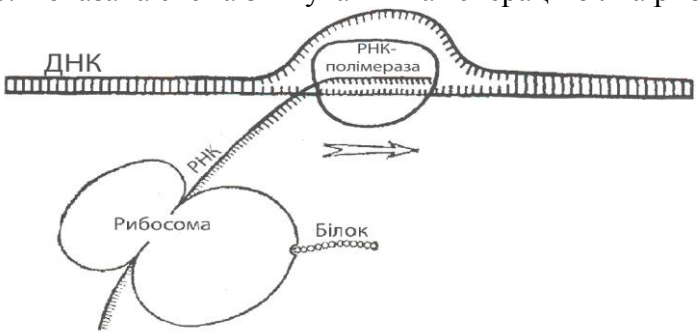


Рис. 18.7. РНК-полімераза повзе по ДНК, синтезуючи РНК. Рибосома зчитує інформацію з РНК синтезуючи білок, у відповідності з генетичним кодом.

Важливою властивістю рибосом є рекурентна перевірка правильності послідовності нуклеотидів в РНК, можливість їх виправлення або припинення генерації білків при наявності невиправних помилок.

Біологічні дослідження показують, що рибосома аналізує не більше 10-12 нуклеотидів. Тобто ключ рекурентності у принципі не повинен перевищувати цю розрядність.

Подібні характеристики мають так звані «липці кінці» ДНК (наприклад, 12 нуклеотидів ДНК λ-фага (рис.18.8).

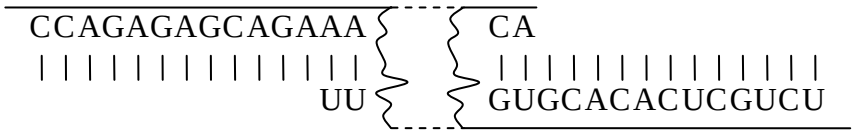


Рис. 18.8. Липкі комплементарні кінці ДНК λ-фага.

Доведено також, що ДНК закручена у спіраль з циклом 10 нуклеотидів і може у хромосомах закручуватись у зверхспіралі, з можливістю ідентифікації промоторів генів у зверх спіральному, спіральному та лінійному станах.

Генетиками доказано, що число нуклеотидів A, U, C, G в ДНК завжди однакове і рівне. Причому, A-U легше розірвати енергетично ніж G-C. У послідовностях ДНК біологічних видів відсутні повторення однакових нуклеотидів більше 32-х, що обмежує можливість генерації ДНК довжиною більше $M=4^{32}$. Особливістю кодування нуклеотидних послідовностей ДНК є трьохфазна синхронізація старт-стопних кодонів у прямому та зворотньому напрямках зчитування інформації (наприклад рис. 18.9).

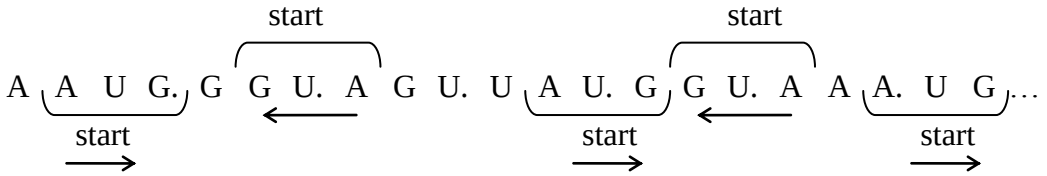


Рис.18.9. Можливе фазування стартових кодонів ДНК.

У процесі мейозу, тобто злиття розкручених замкнутих у кільце форм РНК відбувається їх зближення і прокручування до моменту повного комплементарного злиття і утворення ДНК нового живого організму

18.3. Теорія лінійно-просторових структур кодів поля Галуа.

Викладений короткий аналіз історії відкриття ДНК та особливостей її нуклеотидних послідовностей дозволяє зробити наступні спостереження відносно подібності кодів ДНК та кодів поля Галуа, які є фундаментальною основою інформаційної теорії кодування нуклеотидних послідовностей ДНК.

Гіпотеза про існування форм кодування генетичної інформації у біологічних системах на основі теорії кодів поля Галуа вперше викладена автором у 1991 році у додатку до докторської дисертації, де констатується, що «що для кодування послідовності нуклеотидів ДНК достатньо представлення їх послідовності у полі $GF\binom{R}{4}$, де R довжина кодового

ключа. Наприклад позначивши A=0, T=3, G=1, c=2 можна синтезувати програмним шляхом неповторимі ланцюги …AATGCCACCG… будь якої довжини. При цьому згортка ДНК у спіраль може відповідати циліндричному представленню КПГ. Оскільки ДНК представлена кодоном з трьох нуклеотидів з кроком спіралі рівному 10-ти, то програмно синтезована модель ДНК із властивостями циліндричних КПГ може бути реалізована при довжині ключа R=30, який відповідає добутку модулів $2\cdot3\cdot5$ першої досконалої форми СЗК.

Хочу зауважити, що питанням створення інформаційної теорії ДНК я почав займатися у 1995 році працюючи у Івано-франківському державному університеті нафти і газу, будучи керівником держбюджетної науково-дослідної теми: «Розробка теорії лінійно-просторових структур кодів поля Галуа та їх застосування для розв'язання технічних та мікробіологічних проблем». Наукова робота під шифром Г-15/93 виконувалась по завданню відділу фундаментальних досліджень Міносвіти та науки України.

Метою роботи було проведення фундаментальних теоретичних та прикладних досліджень по створенню теоретичних основ кодів поля Галуа та розробці інформаційної теорії ДНК.

Робота виконувалася згідно плану:

1. Розробка методів пошуку кодових ключів двійкових, трійкових, та четвіркових кодів поля Галуа.
2. Розробка теорії двомірних циліндрично-спіральних та зверх спіральних кодів Галуа.
3. Дослідження генетичних та білкових структур, як об'єктів інформації.
4. Інформаційна структура гена та оцінка залежності довжини білка від довжини промотора.
5. Властивості «липких кінців» кодів поля Галуа та ДНК.

В цей період опубліковано ряд наукових праць по проблемі інформаційної теорії ДНК:

1. Николайчук Я.М. Теоретичні підходи до проблем формування та генерування кодів ДНК // Матеріали НТК професорсько-викладацького складу ІФДТУНГ.-1995 с. 46-47.

2. Николайчук Я. М., Мельничук С.І. Можливості використання полів Галуа для декодування нуклеотидних послідовностей //Тези НТК професорсько-викладацького складу ІФДТУНГ.-1995 с. 51-53.

3. Николайчук Я.М., Руда Г. Ю. математичні закономірності високоенергетичних форма макромолекул біоенергетики // Методи та приклади контролю якості. - №2, 1998.с.- 74-77.

Аналіз інформаційних характеристик відомих білків по опублікованих даних дозволив встановити аналітичну залежність між

кількістю нуклеотидів у промоторі та кількістю нуклеотидів у генетичному коді відповідних білків у вигляді формули

$$n = 3\hat{\mathrm{E}}\left[\log_2 \frac{N}{3}\right], \qquad (18.5)$$

що відповідає виразам

$$\hat{\mathrm{E}}\left[\frac{N}{3}\right] = 2^{\frac{n}{3}}, \text{ або } n = \frac{\hat{E}\left[\log_2 \frac{N}{3}\right]}{3}, \; n = \frac{\hat{E}\left[\log_2 \frac{24}{3}\right]}{3} = 1,$$

де n – довжина промотора (кількість нуклеотидів);

$\hat{\mathrm{E}}[\bullet]$ – цілочисельна функція заокруглення до більшого цілого;

N – довжина гена (число нуклеотидів).

Останній вираз означає, що структура коду поля Галуа G_4^3 і відповідна структура триплету кодону ДНК відповідають промотору *n=1*, що повинно відповідати коду найпростішої стійкої біологічної структури – амінокислоти.

На рис.18.10 показано існуюче кластерування окремих груп білкових структур з однаковими функціональними властивостями.

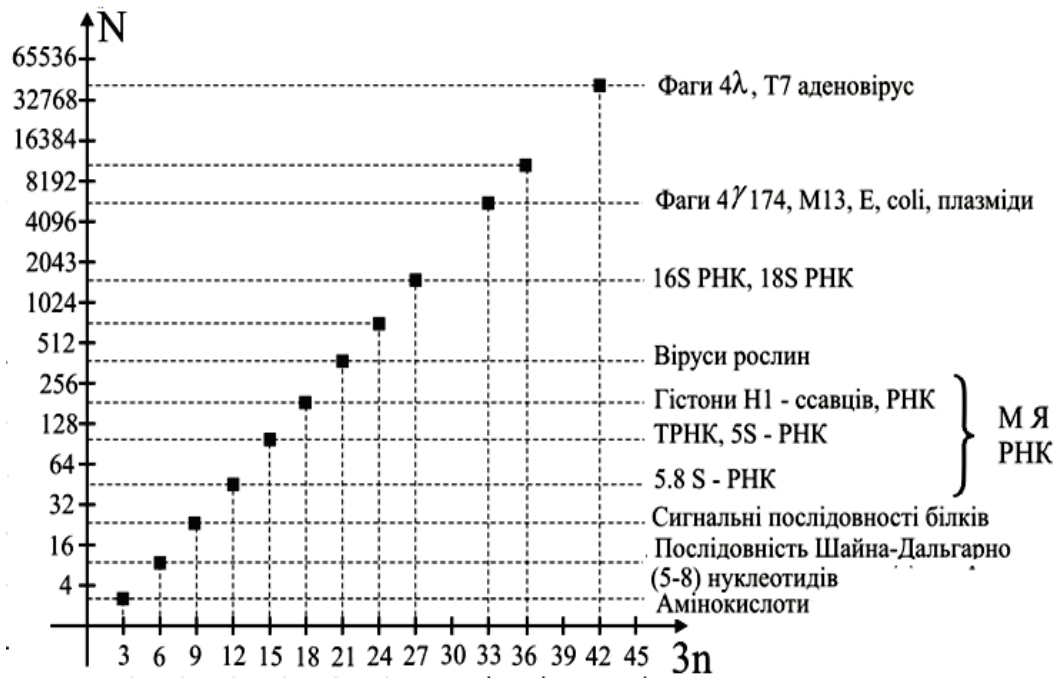


Рис.18.10. Залежність між довжинами промотора та білка у біологічних структурах.

В результаті проведених досліджень автором встановлено ряд подібностей особливих властивостей кодів поля Галуа та ДНК (табл.18.3).

Таблиця 18.3

Порівняння особливих властивостей кодів поля Галуа та ДНК.

№	Особлива властивість	Код поля Галуа	ДНК
1	Ланцюгова взаємозв'язаність	+	+
2	Рекурентність	+	+
3	Циліндрична форма	+	+
4	Можливість виправлення помилок	+	+
5	Можливість «розрізання» та однозначного відновлення при «склеюванні»	+	+
6	Наявність «липких» кінці	+	+
7	Зверхспіралізація без втрати рекурентних властивостей	+	+
8	Наявність кодового ключа рекурентності	+	+

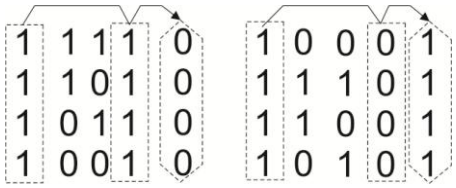
продовження таблиці 18.3

9	«Шеннонівська віддаль» параметрів ключа (промотора) і довжиною коду Галуа	+	+
10	Утворення кільцевої структури з'єднання «липких» кінців	+	+
11	Можлива єдність ключа рекурентності КПГ та рибосоми, яка генерує білки	+	+?

18.4. Метод пошуку кодових ключів послідовностей Галуа та їх інтерпретація в полі $GF\binom{R}{4}$ в нуклеотидних послідовностях ДНК.

Інформаційна технологія пошуку ключів кодових послідовностей Галуа базується на принципах ідентифікації унітарних стовбців фазованих матриць утворених на основі послідовностей кодів Галуа. Наприклад, для $GF\binom{2}{4}$ кодова послідовність має вид (111101011001000). Вибираючи з цієї послідовності кодони ($n+1$) розрядності, які мають ідентичні стартові біти отримуємо наступні фазовані матриці:

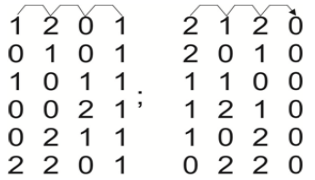
ключ mod 2 ключ mod 2



Звідки однозначно визначається ключ генератора коду Галуа, що відповідає $G_{i+1} = G_{i+1} \oplus G_{i-4}$.

Аналогічно визначається більш складний ключ коду Галуа $GF(^3_3)$, який представляється послідовністю (222120101100211210202201) і породжує фазовані матриці:

ключ mod 3 ключ mod 3



Рішення задачі пошуку ключа Галуа потребує розв'язання системи Діофантових рівнянь:

$(1 \cdot x_1 + 2 \cdot x_2 + 0 \cdot x_3) \bmod 3 = 1;$

$(0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3) \bmod 3 = 1; \quad x_2 = 1;$

$(0 \cdot x_1 + 0 \cdot x_2 + 2 \cdot x_3) \bmod 3 = 1; \quad x_3 = 1;$

$(1 \cdot x_1 + 1 \cdot 2 + 0 \cdot x_3) \bmod 3 = 1; \qquad x_1 = 2.$

Таким чином, ключ Галуа для $GF(^3_3)$ наступний:

$$G_{i+1} = (2G_i + G_{i-1} + 2G_{i-2}) \bmod 3,$$

згідно якого виконуються усі умови рівності фазованих матриць mod 3.

ДНК–подібні коди поля Галуа формуються по mod 4. Наприклад: $GF(^4_4)$ згідно ключа:

$$G_{i+1} = (G_i + G_{i-4}) \bmod 4,$$

формує рекурентну кодову послідовність,

3 3 3 3 2 1 0 3 1 2 2 1 2 0 2 3 1 1 3 2 3 0 3 1 0 0 3 0 0 0.

Позначивши 0-U; 1-C; 2-G; 3-A отримаємо ДНК подібний четвірковий код Галуа довжиною 25 пар комплементарних нуклеотидів з «липкими кінцями», які дозволяють замкнути його в кільце (рис.18.11).

AGCUACGGCGUGACCAGAUACUUAUUU
| | | | | | | | | | | | | | | | | | | | | | | | | |
AAAUCGAUGCCGCACUGGUCUAUGAAU

Рис.18.11. ДНК-подібний код Галуа.

Використовуючи табл.18.4 кодонів амінокислот можна зобразити модель ДНК у вигляді послідовності амінокислот у наступному вигляді (рис.18.12).

Таблиця 18.4.

Коди амінокислот ДНК.

A	*A*	$A \vee G \Rightarrow ЛІЗ$	$U \vee C \Rightarrow АСН$	
A	*U*	$G \Rightarrow МЕТ$	$A \vee U \vee C \Rightarrow ІЛЕ$	
A	*C*	$A \vee U \vee C \vee G \Rightarrow ТРЕ$		
A	*G*	$A \vee G \Rightarrow АРГ$	$U \vee C \Rightarrow ТЕР$	
U	*A*	$A \vee G \Rightarrow ТЕР$	$U \vee C \Rightarrow ТІР$	
U	*U*	$A \vee G \Rightarrow ЛЕЙ$	$U \vee C \Rightarrow ФЕН$	
U	*C*	$A \vee U \vee C \vee G \Rightarrow СЕР$		
U	*G*	$A \Rightarrow ТЕР$	$U \vee C \Rightarrow ЦІС$	$G \Rightarrow ТРІ$
C	*A*	$A \vee G \Rightarrow ГЛН$	$U \vee C \Rightarrow ГІС$	
C	*U*	$A \vee U \vee C \vee G \Rightarrow ЛЕЙ$		

продовження таблиці 18.4

C	*C*	$A \vee U \vee C \vee G \Rightarrow ПРО$	
C	*G*	$A \vee U \vee C \vee G \Rightarrow АРГ$	
G	*A*	$A \vee G \Rightarrow ГЛУ$	$U \vee C \Rightarrow АСП$
G	*U*	$A \vee U \vee C \vee G \Rightarrow ВАЛ$	
G	*C*	$A \vee U \vee C \vee G \Rightarrow АЛА$	
G	*G*	$A \vee U \vee C \vee G \Rightarrow ГЛІ$	

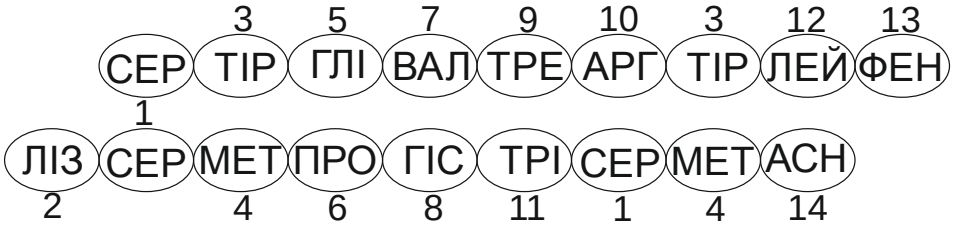


Рис. 18.12. Ідентифікація триплетів кодонів нуклеотидів ДНК коду Галуа в полі $GF(\binom{4}{4})$.

Якщо зчитувати з кодової послідовності $GF(4^4)$ коди амінокислоти з кроком одного нуклеотида, то отримаємо неповний набір кодонів амінокислот.

1. ААА-СЕР	5. UAC-ТІР	7. CGU-АРГ	13.ССА-ПРО	5. UAC-ТІР
2. ААG-ЛІЗ	6. ACG-ТРЕ	10.GUG-ВАЛ	14. САG-ГЛН	6. ACU-ТРЕ
1. AGC-СЕР	7. CGG-АРГ	11.UGA-ТЕР	7. AGA-АПГ	4.CUU-ЛЕЙ
3. GCU-АЛА	8. GGC-ГЛІ	12.GAC-АСП	12.GAU-АСП	4.UUA-ЛЕЙ
4. CUA-ЛЕЙ	9. ПСП-АЛА	6. АСС-ТРЕ	15.AUA-ІЛЕ	5.UAU-ТІР

Розглянемо простіший приклад.

Нехай стартовий кодон поля Галуа $GF(4^3)$ має четвірковий код (3 3 3) тоді згідно ключа $G_{i+1} = (G_i + G_{i-2}) \bmod 4$ отримаємо комплементарну рекурентну послідовність:

3 3 3 2 1 0 2 3 3 1 0 3 3 0 0
| | | | | | | | | | | | | | |
0 0 0 1 2 3 1 0 0 2 3 0 0 3 3

.

Заміною $3 \to A$, $1 \to G$, $2 \to C$, $0 \to U$ побудуємо ДНК – подібний код Галуа (рис.18.13) “липкими” кінцями і відповідними можливими кодонами амінокислотами при його зчитування у різних напрямках з різними фазами.

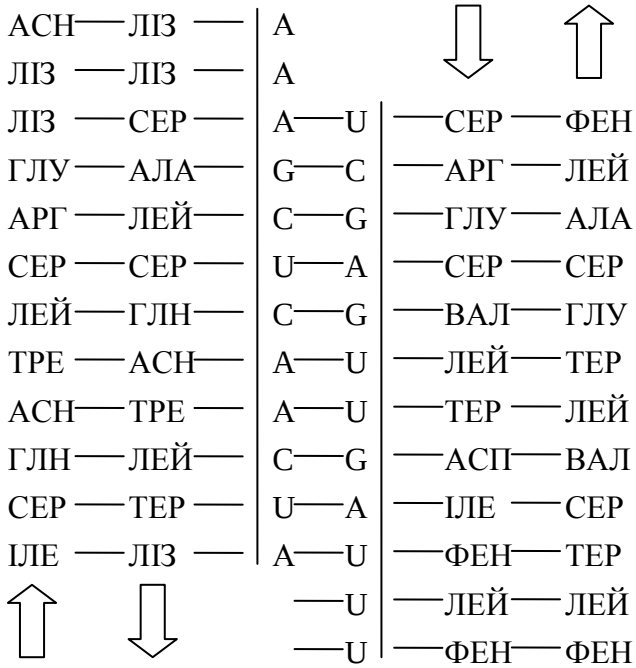
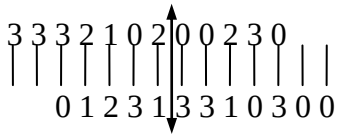


Рис.18.13. ДНК–подібний код Галуа поля $GF(^{3}_{4})$.

Аналіз приведених прикладів генерації ДНК – подібних кодів поля Галуа з простими ключами по модулю 4 показує, що у першому випадку в полі $GF(^{4}_{4})$ присутні 12 кодонів амінокислот і відсутні (6)

У даному випадку у полі $GF(^{4}_{3})$ відсутні кодони амінокислот (ТІР, ЦІС, ТРІ, ПРО, ГІС, МЕТ, ГЛІ, ТІР), що свідчить про недосконаліть, неправильність або невірність визначення ключа Галуа, який не відповідає умовам комплементарності всіх типів нуклеотидів та кодонів амінокислот ДНК.

Відомо, що ДНК закручена у подвійну спіраль з властивостями спіралі Мебіуса. Таким чином, якщо виконати поворот у кінці послідовності коду поля Галуа $GF(^{4}_{3})$ по Мебіусу:



отримаємо лінійну нову структуру

$$3332102331030001231002 30,$$

що відповідає ДНК-подібному комплементарному коду Галуа (рис.18.14).

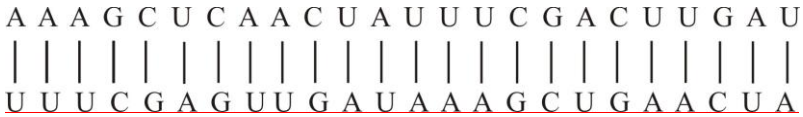


Рис.18.14. Мебіусний ДНК-подібний комплементарний код Галуа.

Одержана послідовність коду Галуа за рахунок склеювання коду GF_{3}^{4} по Мебіусу знімає протиріччя обмеженості довжини послідовності Галуа за рахунок вираженості та парності mod 4. При цьому виконується умова рівняння залежності довжини гена від довжини промотора (18.5).

18.5. Характеристики компліментарності нуклеотидів ДНК.

Ступінь комплементарної взаємодії, тобто притягування (+), відштовхування (-) чи відсутності взаємодії (0) визначається на молекулярному рівні між структурами молекул, які позначені символами нуклеотидів A, U, T, G, C.

На рис.18.15 приведені формалізовані структури молекул нуклеотидів A, U, T, G, C, які утворюють послідовність ДНК та РНК.

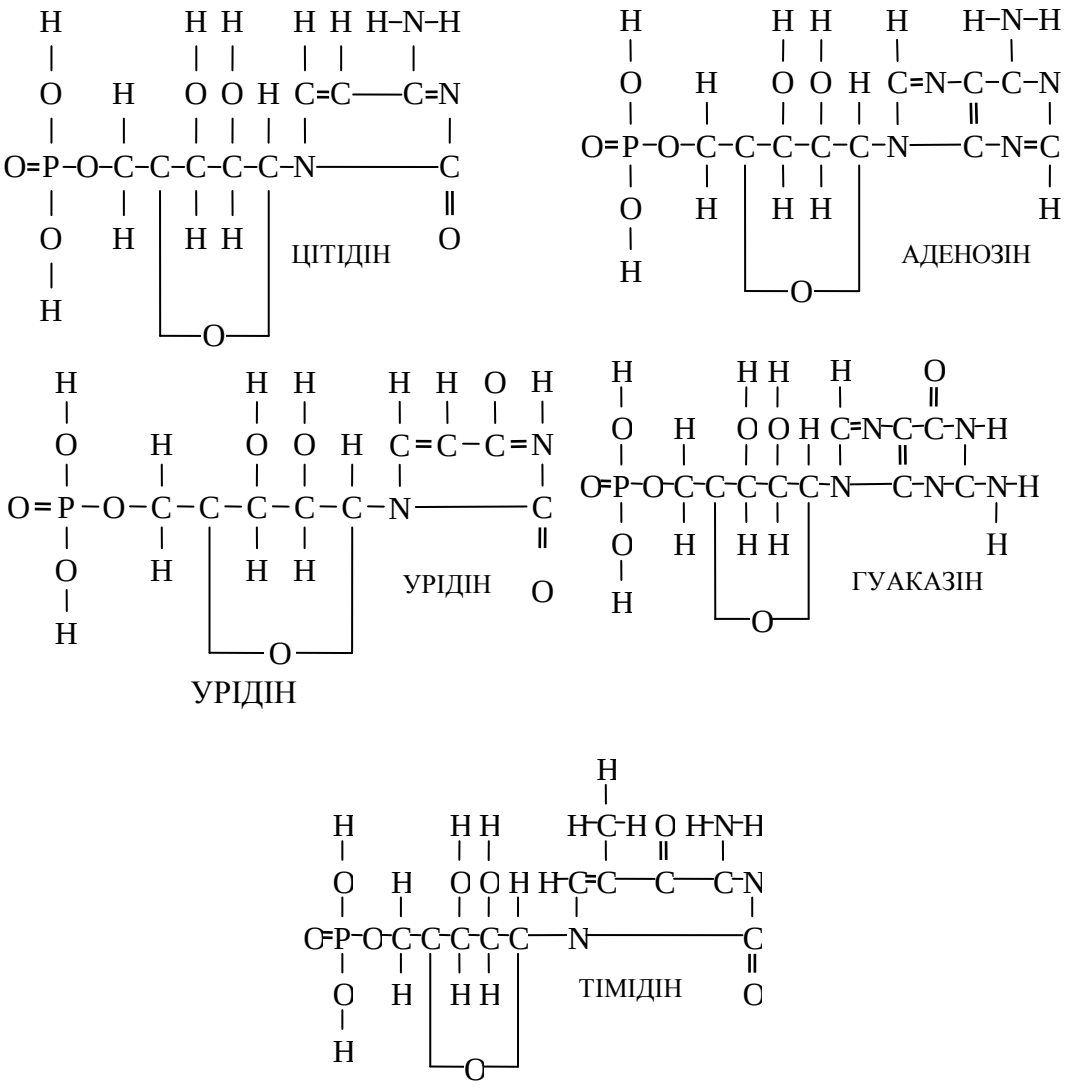


Рис.18.15. Формалізовані структури молекул нуклеотидів A, U, T, G, C.

Таким чином процес генерації ДНК–подібних кодів Галуа у загальному випадку можна представити моделлю-графом комплементарної взаємодії моно фосфатних нуклеотидів A, U, T, G, C (рис.18.16).

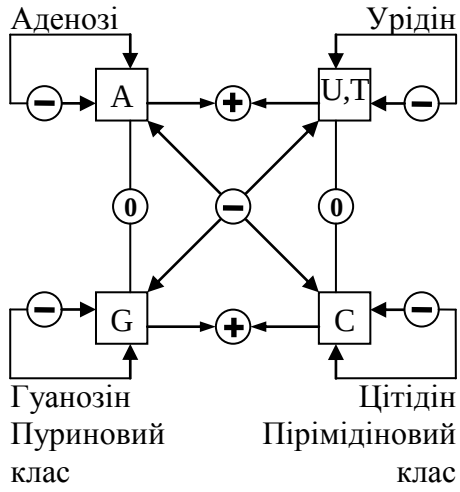


Рис.18.16. Модель графа комплементарних взаємодій молекул нуклеотидів ДНК.

На рис.18.16 стрілками та кружечками показаний характер взаємодії нуклеотидів ДНК притягання (→⊕←), відштовхування (←⊖→) та нейтральної взаємодії (—⓪—) окремих пар нуклеотидів ДНК, що демонструється також наступною таблицею:

	A	U,T	C	G
A	-	+	0	-
U,T	+	-	-	-
C	0	-	-	+
G	-	-	+	-

Базуючись на теорії квантової механіки та молекулярної фізики можна конкретизувати число значення комплементарної взаємодії притягання (→⊕←), відштовхування (←⊖→) та інваріантності (—⊙—) молекул A, U, T, G, C, які утворюють послідовність ДНК та РНК (рис.18.17).

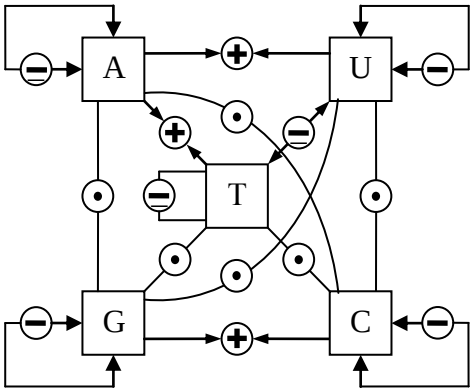
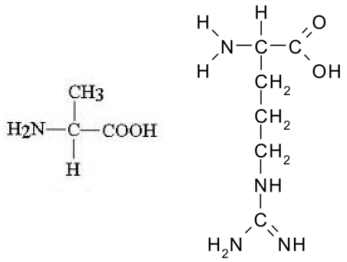


Рис.18.17. Граф взаємодії пар нуклеотидів ДНК та РНК.

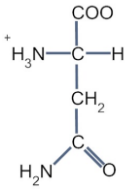
M. Singer та P. Berg приводять приклад комплементарної взаємодії нуклеотидів T–A і C–G у вигляді водневих зв'язків на міжатомних віддалях, що характеризує практично два рази слабший (+) зв'язок у парі A↔U,T по відношенню до пари C–G. Утворення пар між двома пуринами, двома пірімідінями чи не комплементарними нуклеотидами (A– C або G– U) теоретично утруднене, оскільки при цьому не можуть утворюватись водневі зв'язки і відповідно порушується геометрія спіралі ДНК.

На рис.18.18 приведені структурні формули амінокислот, символічні буквені (A, R, N, D, C, Q, E, G, H, I, K, L, M, F, P, S, T, W, Y, V) та повні назви відомих амінокислот, які утворюють спіральні комплементарні ланцюги ДНК.

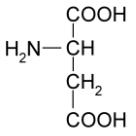


Аланін (A; Ala)

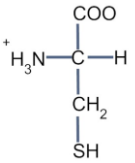
Аргінін (R; Arg)



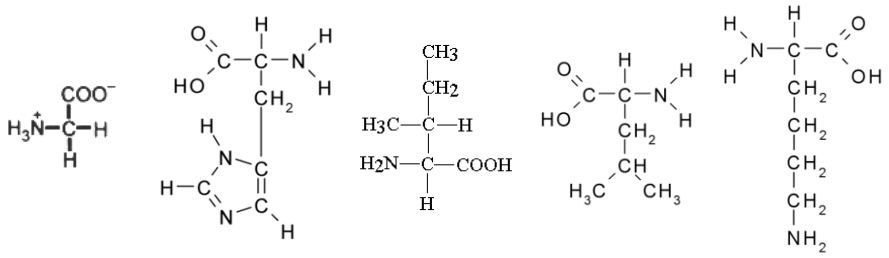
Аспарагін (N; Asn)



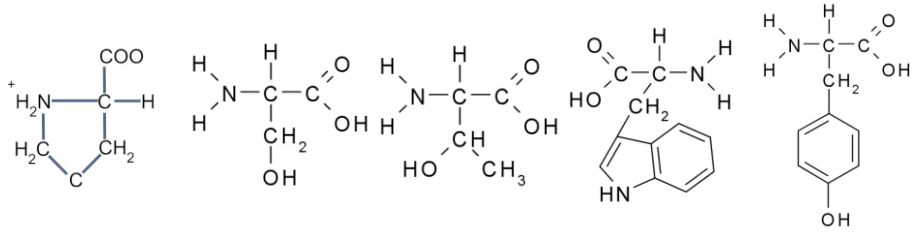
Аспарат (D; Asp)



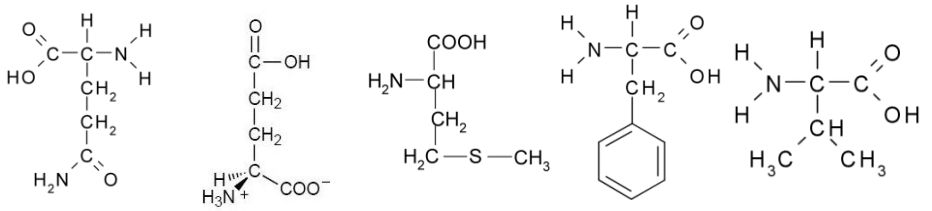
Цістеін (C; Cys)



Гліцин (G; Gly) | Гістідін (H; His) | Ізолейцин (I; IIe) | Лейцин (L; Leu) | Лізін (K; Lys)



Пролін (P; Pro) | Серін (S; Ser) | Треонін (T; Tre) | Тріптофан (W; Trp) | Тірозін (Y; Tyr)



Глутамін (Q; Gln) | Глутамат (E; Glu) | Метіонін (M; Met) | Фенілаланін (F; Phe) | Валін (V; Val)

Рис.18.18. Структурні формули амінокислот та їх символік.

Gail L Rosen приводить таблицю виконання операцій додавання та множення в полі Галуа та її інтерпретації символу *GF* (4) на основі незвідного полінома $a^2+a+1=0$ та їх інтерпретації у символах нуклеотидів ДНК:

$a=0\leftrightarrow 0\leftrightarrow A$
$a^0=1\leftrightarrow 1\leftrightarrow G$
$a^1=a\leftrightarrow 2\leftrightarrow T$
$a^2=a+1\leftrightarrow 3\leftrightarrow G$

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

В праці S. Robersy, G. Ricardo, R. Mopago досліджується векторний простір надгенетичним кодом ДНК у розширеному полі Галуа $GF(5^3)$ з елементами $x_1\ x_2\ x_3$, де $x_i \in$ (O, A, C, G, U), на основі незвідних поліномів по модулю 5: x^2+3x+4 та $4x^2+3x+2$, та Абелевої групи C_{125}. Також показано, що автоморфізм груп Галуа може бути корисним інструментом для вивчення мутацій у *N*-вимірному векторному просторі всіх можливих ДНК послідовностей. Авторами отримані фазовані матриці послідовностей нуклеотидів біологічних ДНК, які приведені на рис.18.19 і висунута гіпотеза про довжину ключа генерації коду ДНК загальною довжиною 51 нуклеотид.

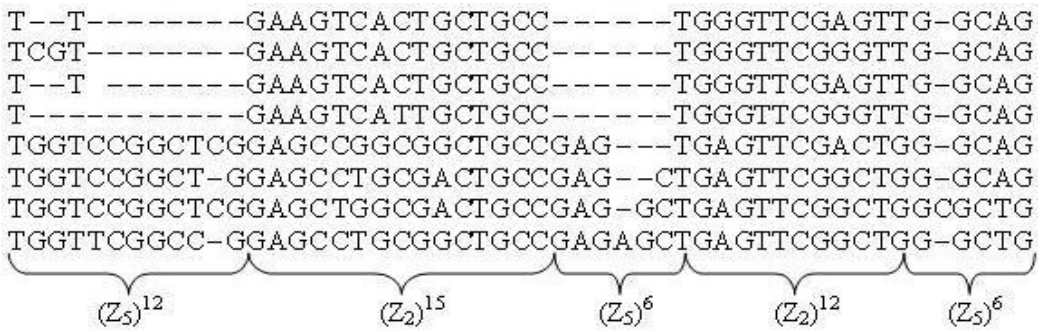


Рис.18.19. Фазовані матриці біологічних ДНК. Приклад вирівнювання групи $S = \left((Z_5)^{12} \oplus (Z_2)^{15} \oplus (Z_5)^6 \oplus (Z_2)^{12} \oplus (Z_5)^6 (\mathrm{mod}\ P)\right)$.

Очевидно, що ключ генерації кодів біологічної ДНК не може перевищувати 12 з врахуванням 10-нуклеотидного кроку її закручування у спіраль. Тобто приведені фазовані матриці відповідають циліндричній формі кодів Галуа з реалізацією ключів по твірних циліндра.

Проблему дослідження співвідношення компліментарності в записах нуклеотидів по одній нитці в хромосомах ДНК та ефективності оптимальних байєсовських процедур розпізнавання вирішували І.В. Сергієнко, О.В. Палагін та А.М. Гупал.

А.М.Гупал досліджував симетрію запису генетичної інформації в ДНК і встановив, що за допомогою моделі ланцюгів Маркова можна легко генерувати випадкові ДНК-подібні послідовності, для яких буде виконуватися симетрія для моделі Уотсона-Кріка. Також відмічені широкі можливості застосування байєсовських процедур на моделях ланцюгів Маркова при рішенні складних задач передбачення просторової структури білків для розпізнавання властивостей участків генів, в тому числі генетичних захворювань.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Теорія чисел, абелевих груп та полів Галуа.

1. Алексич Г. Проблемы сходимости ортогональных рядов / Алексич Г. – М.: Инлитиздат, 1963. – 359 с.
2. Анісімов А.В. Алгоритмічна теорія великих чисел. Модулярна арифметика великих чисел.-К.: Видавничий дім «Академперіодика», 2001.-152с.
3. Артин Э. Теория Галуа. М.: МЦНМО.– 2008.
4. Болотов А.А., и др.. Алгоритмические основы эллиптической криптографии.-М.:МэИ.2006.-111с.
5. Боревич З. И., Шафаревич И. Р. Теория чисел. — М.: Наука, 1972. – 510 с.
6. Боровков А. А. Теория вероятности. – М.: Наука, 1976. –352 с.
7. Бухштаб А.А. Теория чисел / Бухштаб А.А. - М.: Просвещение, 1966. – 384 с.
8. Валях П. Последовательно-параллельные вычисления. - М.: Мир, 1985. - 378с.
9. Вандер Варден Б.Л. Алгебра. - М.: Наука, 1980. - 624 с.
10. Вариченко Л. В. Абстрактные алгебраические системы и цифровая обработка сигналов / Вариченко Л. В., Лабунец В. Г., Раков М. А. – К.: Наук. думка, 1986. – 248 с.
11. Вариченко Л.В. Ортогональные разложения булевых функций над полем комплексных чисел и полями Галуа. - Львов, 1979. - 26 с. Рукопись деп. в ВИНИТИ, № 993. - 79 Деп.
12. Вентцель Е.С. Теория вероятностей. - 'М.: Наука, 1969, 576 с.
13. Вентцель А. Д. Курс теории случайных процессов. – М.: Наука, 1975. – 320 с.
14. Виноградов И. М Основы теории чисел. – М.-Л.: Гостехиздат, 1952. – 180 с.
15. Вулих Б. З. Введение в функциональный анализ / Вулих Б. З. – М.: Наука, ГРФМЛ, 1967. – 416 с.
16. Галуа // Энциклопедический словарь Брокгауза и Ефрона: В 86 томах (82 т. и 4 доп.). – СПб., 1890–1907.
17. Галуа Э. Сочинения. С приложением статьи П. Дюпюи: Жизнь Эвариста Галуа. М.-Л.: Гостехиздат, 1936.
18. Гаусс К.Ф. Труды по теории чисел: Пер. с нем. - М.: АН СССР, 1959. - 978 с.
19. Дальма А. Эварист Галуа: Революционер и математик. М.: Наука, 1984.
20. Ефимов А. В. Математический анализ (специальные разделы) / Ефимов А. В. – М.: Высшая школа, 1980. – 279 с.
21. Задирака В.К., Кудин А.М. Построение программно-аппаратных комплексов арифметики сверхбольших чисел // Комп'ютерна математика. Оптимізація обчислень: Зб. Наук праць / Т.1. – Київ: Ін-т кібернетики ім В.М.Глушкова НАНУ, 2001. – С.158-163.
22. Задирака В.К., Мельникова С.С. Быстрое умножение многорозрядных чисел с использованием БПФ // Кибернетика и системный анализ. – 1996. - №3. – С.63-67.
23. Задирака В.К., Мельникова С.С, Терещенко А.М. Оптимізація алгоритмів швидкого множення великих чисел. Ч.1 // УсіМ. -2006. - №2 – С.23-38; Ч.2. - УсіМ. -2006. - №3 – С.78-94.

24. Инфельд, Л. Эварист Галуа. Избранник богов. М.: Молодая гвардия, 1965, С. 259–260.
25. К. Айерлэнд, М. Роузен Классическое введение в современную теорию чисел. – М.: Мир, 1987.
26. Карацуба А.А. Сложность вычислений//Тр.МИАМ, 1995.Т.211.-С.186-202.
27. Касянчук М. М. Теорія алгоритмів перетворень китайської теореми про залишки в матрично-розмежованому базисі Радемахера-Крестенсона / М. М. Касянчук, Я. М. Николайчук, І. З. Якименко // Вісник національного університету «Львівська політехніка». – 2011. – № 688. – С. 118–124.
28. Касянчук М.М. Теоретичні основи аналітики та алгоритми оптимізації обчислень простих чисел. // М.М.Касянчук, І.З.Якименко, О.І.Волинський, С.В.Івасьєв / Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту // Матеріали Міжнародної проблемно-наукової міжгалузевої конференції «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління (ПНМК-2010) ».-В.6, т.1.- с.33-36.
29. Кох Х.Алгебраическая теория чисел. – М.: ВИНИТИ, 1990. – Т. 62. – 301 с.
30. Лабунец В.Г. Теоретико-числовые преобразования над полями алгебраических чисел. - В кн.: Применение ортогональных методов при обработке сигналов и анализе систем. - Свердловск: УПИ, 1981, с.44-54.
31. Ленг С. Алгебра. - М.: Мир, 1968. - 564 с.
32. Маккелан Дж. Х., Рэйдер Ч.М. Применение теории чисел в цифровой обработке сигналов- М.: Радио и связь, 1983. - 264 с.
33. Манин Ю.И., Панчишкин А.А. Введение в теорию чисел. – М.: ВИНИТИ, 1990. – Т. 49. – 341 с.
34. Николайчук Я.Н. Применение методов теории чисел для сжатия измерительной информации в системах телеконтроля процессов бурения / Я.Н. Николайчук, В.П. Божнев, С.Я. Зевелев // Материалы Всесоюзной конференции молодых ученых нефтяных ВУЗов. - М.:МИНХиГП, 1975. – С. 134-138.
35. Постников М. М. Теория Галуа. М.: Наука, 1963, 517.– П 63.– 220 с.
36. Саймон Сингх. Великая теорема Ферма. Перевод с английского Ю. А. Данилова. — М.: МЦНМО, 2000 — ISBN 5-900916-61-8.
37. Сизый С.В. Лекции по теории чисел. — Екатеринбург: Уральский государственный университет им. А. М. Горького, 1999.
38. Соловьев Ю. Эварист Галуа, Квант 1986 год, номер 12.
39. Фукс Б.А., Шабат Б.В. Функции комплексного переменного и некоторые их приложения. - М.: Физматиздат, 1959. - 375 с.
40. Функциональный анализ / [под общ. ред. С. Г. Крейна]. – [2-е изд.]. – М.: Наука, 1972. – 544 с.
41. Хинчин А. Я. Три жемчужины теории чисел. – М.: Наука, 1979. ¬– 64 с.
42. Чеботарёв Н.Г. Основы теории Галуа. Часть 1 / Чеботарёв Н.Г. – М.: ОНТИ, 1934. – 314 с.
43. Чеботарёв Н.Г. Основы теории Галуа. Часть 2 / Чеботарёв Н.Г. – М.: ОНТИ, 1937. – 155 с.

Теоретико числові базиси.

1. Агарвал Р., Баррас С. Теоретико-числовые преобразования для быстрого вычисления цифровой свертки. - ТИИЭР, 1975, вып.4, С. 4-20.
2. Агарвал Р.С., Кули Дж.У. Новые алгоритмы для цифровой свертки. - В кн.: Применение теории чисел в цифровой обработке сигналов. - М.: Радио и связь, 1983, с.91-117.
3. Азов А.К., Ожиганов А.А. О преобразовании псевдослучайных кодов / Функциональная оптоэлектроника в вычислительной технике и устройствах управления. - Тбилиси, 1986. - 141 с.
4. Айфичер Э. С. Цифровая обработка сигналов: практический подход / Э. С. Айфичер, Б. У. Джервис; пер. с англ. И. Ю. Дорошенко, А. В. Назаренко. – [2-е изд.]. – М.: Издательский дом “Вильямс”, 2004. – 992 с.
5. Акушский И.Я., Амербаев В.М., Пак И.Т. Основы машинной арифметики комплексных чисел. - Алма-Ата: Наука, 1970.- 248 с.
6. Акушский И.Я. Машинная арифметика в остаточных класах / И.Я. Акушский, Д.И. Юдицкий – М.: Сов. радио, 1978. – 256 с.
7. Ахмед Н. Ортогональные преобразования при обработке цифровых сигналов / Н. Ахмед, К. Р. Рао; пер. с англ. Т. Э. Кренкеля. – М.: Связь, 1980. – 248 с.
8. Ахмед Н., Рао К. Ортогональные преобразования при обработке цифровых сигналов. - М.: Связь, 1980. - 247 с.
9. Ахо Альфред, В. Хопкрофт, Джон, Ульман, Джеффри Структуры данных и алгоритмы.: Пер.с англ.: М.: Издательский дом “Вильямс”, 2001.- 384с.
10. Ачасова С. М., Бандман О. Л. Корректность параллельных вычислительных процессов. – Новосибирск: Наука. Сиб. отд-ние, 1990. – 253 с.
11. Блейхут Р. Быстрые алгоритмы цифровой обработки сиг налов: Пер. с англ. - М.: Мир, 1989. - 448 с.
12. Гулямов С.С., Зайнидинов Х.Н. Синтез параллельно-конвейерных вычислительных структур для выполнения быстрых преобразований Хаара.// Методы и модели систем обработки. - Сб. научн. трудов НПО "Кибернетика" АН РУЗ. Ташкент. -1994. - С.124-127.
13. Дагман Э. Г., Кухарев Г. А. Быстрые дискретные ортогональные преобразования. Новосибирск, 1983. – 284с.
14. Добеши И. Десять лекций по вейвлетам / Ингрид Добеши. – Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2001. – 464 с.
15. Дядюнов Н.Г. Ортогональные и квазиортогональные сигналы / Н. Г. Дядюнов, А.И.Сенин. – М.: Связь, 1977. – 222 с.
16. Задірака В.К. Теоретичні основи та високопродуктивний алгоритм обчислення мультистепеневої функції в базисі Крестенсона. // В.К.Задірака, Я.М.Николайчук, Касянчук М.М. / Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту // Матеріали Міжнародної проблемно-наукової міжгалузевої конференції «Інформаційні проблеми комп’ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління (ПНМК-2010) ».-В.6, т.1.- с.30-32.

17. Задирака В.К. Анализ сложности алгоритма умножения сверхбольших чисел на основе коэффициентов Уолша // Кибернетика и системный анализ. – 2001. - №6. – С.99-110.
18. Залманзон Л. А. Преобразования Фурье, Уолша, Хаара и их применение в управлении связи и других областях / Залманзон Л. А. – М.: Наука. Гл. ред. физ.-мат. лит., 1989. – 496 с.
19. Заставний О.М. Дослідження теоретико–числових базисів як основи побудови двомірних шумоподібних сигналів // Вісник Національного університету «Львівська політехніка» «Радіоелектроніка та телекомунікації».-2004 – №508. С.33-37.
20. Канторович Л.В. Функциональный анализ в нормированных пространствах / Канторович Л.В., Акилов Г.П. – М.: Гос. изд. физ.-мат. лит., 1959. – 684 с.
21. Карповский М. Г. Спектральные методы анализа и синтеза дискретных устройств / М. Г. Карповский, Э. С. Москалев – Л.: Энергия, 1973. – 144 с.
22. Касами Т. и др. Теория кодирования. - М.: Мир, 1978.-576.
23. Касянчук М. М. Теорія алгоритмів RSA та Ель-Гамаля в розмежованій системі числення Радемахера-Крестенсона// М. М. Касянчук, І. З. Якименко, О. І. Волинський, І.Р. Пітух / Вісник Хмельницького національного університету. – 2011. – № 3. – С. 265–273.
24. Касянчук М. М. Теорія алгоритмів пошуку найбільшого спільного дільника у базисі Крестенсона / Касянчук М., Якименко І., Николайчук Я. // Вісник ТНТУ. — 2011. – Том 16. – № 1. – С. 154–161.
25. Касянчук М.М. Концепція теоретичних положень досконалої форми перетворення Крестенсона та його практичне застосування. // М.М.Касянчук / Міжнародний науково-технічний журнал «Оптико-електронні інформаційно-енергетичні технології». - №2 (20). – 2010. – с.43-47
26. Касянчук М.М. Теорія та оптимізація алгоритмів опрацювання великорозрядних чисел у базисі Крестенсона.// Касянчук М.М., Якименко І.З., Івасьєв С.В. / Міжнародна молодіжна математична школа «Питання оптимізації обчислень (ПОО- XXXVII), Україна, Крим Велика Ялта, смт.Кацивелі, 22-29 вересня 2011 року.
27. Колмогоров А. Н., Фомин С. В. Элементы теории функции и функционального анализа. –М.: Наука, 1976. – 352 с.
28. Корнилов А.И., Семенов М.Ю., Ласточкин О.В. Принципы построения модулярных индексных умножителей // Известия ВУЗов. Электроника. – 2004. - №2. – С.48-55.
29. Лабунец В.Г. Обобщенные преобразования Хаара. - В кн.: Многозначные элементы, структуры, системы. - Киев: Наукова думка, 1983, с.78-85.
30. Мирский Г.Я. Аппаратурное определение характеристик случайных процессов. - М.: Энергия, 1972. - 456 с.
31. Николайчук Я. М. Теорія цифрових перетворень мультибазисного супершвидкодіючого процесора / Я. М. Николайчук // Штучний інтелект. – 2008. – № 4. – С. 387 – 394.
32. Николайчук Я., Сегін А., Сабадаш І. Метод розпізнавання графічних зображень на основі власних функцій Карунена-Лоєва. // Комп'ютерні технології друкарства.

Збірник наукових праць. – Львів, – № 5, – 2000. – С. 344–347.
33. Николайчук Я.М. Теоретичні основи побудови спецпроцесорів у базисі Крестенсона / Я.М. Николайчук, О.І. Волинський, С.В. Кулина // Вісник Хмельницького національного університету. – 2007. – Т.1, №3. – С. 85-90.
34. Николайчук Я.М. Теорія цифрових перетворень мультибазисного супершвидкодіючого процесора. Научно-теоретический журнал "Искусственный интеллект". ІПШІ МОН і НАН України "Наука і освіта". – 2008. - №4. – С.387-394.
35. Николайчук Я.М., Теоретико-числові базиси Крестенсона та Галуа – фундаментальна основа оптимізації опрацювання велико розрядних чисел. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2011 - №7.-С.114-122.
36. Николайчук Я.Н. Метод уплотнения информации, вводимой в ЭВМ / Я.Н. Николайчук, В.П. Божнев // Управляющие системы и машины. – 1977. - №1. – С. 68-74.
37. Николайчук Я.Н. О представлении информации в многоканальных информационно-измерительных системах. Респ. сб. «Изменение, контроль и автоматизация в нефтяной и газовой промышленности». – К.: “Техніка”, 1974. – С.24-27.
38. Николайчук Я.Н. Представление измерительной информации в нормализованной системе исчисления остаточних классов / Я.Н. Николайчук, З.Н. Крикун, В.П. Божнев // Известия ВУЗов “Нефть и газ”, 1976. - №6. – С. 63-70.
39. Николайчук Я.М. Напрямки розвитку процесорів комп’ютерних систем на основі дискретних теоретико-числових базисів. Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2008. - №4. Т1. – С.8-11.
40. Оре О. Графы и их применение.– М.: Мир, 1965.–173с.
41. Петришин Л. Б. Теоретичні основи перетворення форми та цифрової обробки інформації в базисі Галуа: [навчальний посібник] / Петришин Л. Б. – К.: ІзіМН МОУ, 1997. – 237 с.
42. Превисокова Н.В., Петришин Л.Б. Теоретико-числові основи дискретного гармонічного аналізу в системі Рademахера // Вісник Прикарпатського університету. Математика. Фізика. – 2007. – Вип. 3. – С. 107–112.
43. Свердлик М. Б. Оптимальные дискретные сигналы. М.: Сов. радио, 1975. - 200 с.
44. Соболь И. М. Многомерные квадратурные формулы и функции Хаара. – М.: Наука, 1969. – 288 с.
45. Соучек Б. Мини-ЭВМ в системах обработки информации. - М.: Мир, 1976. - 520 с.
46. Трахтман А.М., Трахтман В.А. Основы теории дискретных сигналов на конечных интервалах. - М.: Сов.радио, 1979. - 208 с.
47. Хуанг Т.С. и др. Быстрые алгоритмы в цифровой обработке изображений / Пер. с англ. - М.: Радио и связь, 1984. - 224 с.
48. Яблонский В. С. Введение в дискретную математику. – М: Наука, 1986. – 384с.

49. Яглом А. М., Яглом И. М. Вероятность и информация. Главная редакция физико-математической литературы издательства «Наука», 1973. – 512 с.
50. 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.

Фундаментальні положення кібернетики та теорії інформаційних систем.

1. Акушский И.Я., Амербаев В.М., Пак И.Т. Основы машинной арифметики комплексных чисел. - Алма-Ата: Наука, 1970.- 248 с.
2. Акушский И.Я. Машинная арифметика в остаточных класах / И.Я. Акушский, Д.И. Юдицкий – М.: Сов. радио, 1978. – 256 с.
3. Алексеев В.Е. Основы информационных технологий. Графы и алгоритмы. Структуры данных. Модели вычислений.: М.: BHV, 2006. – 320 с.
4. Алишов Н.И. Развитые методы взаимодействия ресурсов в распределенных системах / Алишов Н.И. – К.: Сталь, 2009. – 449 с.
5. Боюн В. П. Теоретико-информационные основы преобразования и обработки информации в системах реального времени // Проектирование и применение средств микропроцессорной техники: Сб. научн. Трудов.– Киев: ИК им. В.М.Глушкова АН УССР, 1986. – С. 9–17.
6. Боюн В. П. Інформаційні аспекти інтелектуального сприйняття візуальної інформації в системах технічного зору. Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2011. - №7. Т1. – С.27-32
7. Боюн В.П. Динамическая теория информации. Основы и приложения – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2001. – 326 с.
8. Боюн В.П. Торетико-информационные основы преобразования и обработки информации в системах реального времени // Проектирование и применение средств микропроцессорной техники:Сб. научн.трудов. - Киев: ИК им.В.М.Глушкова АН УССР, 1986. - С.9-17.
9. Боюн В.П. Методы определения δ-энтропии случайных процессов // УСиМ. – 2000. –№ 4. – С.14-19.
10. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и моделирование мультипроцессорных систем.- К.: Наукова думка, 1990, 256с.
11. Вилкас Э. Й. Решения: теория, информация, моделирование. – М.: Радио и связь, 1981. – 328 с.
12. Винер Н. Кибернетика или управление и связь в животном и машине. 2-е издание. – М.: Наука. 1983. – 344с.
13. Виноградов В.И. Информационно-вычислительные системы: Распределенные модельные системы автоматизации. - М. : Энергоатомиздат, 1966. – 336 с.
14. Введение в кибернетическую технику: Обработка физической информации. / Под общ. Ред. Б. Н. Малиновского. – К.: Наук. думка, 1979. – 256 с.

15. Винцюк Т.К. Структура процессоров предварительной обработки и распознавания речевых сигналов / Тезисы докладов школы-семинара "Распараллеливание обработки информации". - Львов; ФМИ, 1989, часть 2, с.109-110.
16. Волкова В.Н., Денисов А.А. Основы теории систем и системного анализа: Учебник, издание 2. – СПб.: Изд-во СПбГТУ, 1999, 256с.
17. Гаврилов М.В. Информатика и информационные технологии.: СПб.: ГАРДАРИКА, 2006.-453 с.
18. Глушков В.М. Основы безбумажной информатики. М.: Наука, 1987. – 552с.
19. Глушков В.М., Иванов В.В., Яненко В.М. Моделирование развивающихся систем. – М.: Наука, 1983.- 352 с.
20. Голдман С. Теория информации.М.: Изд-во иностранной литературы, 1975. 382с.
21. Гоулд Б., Рейдер У. Цифровая обработка сигналов.- М.:Сов.радио, 1973. - 368с.
22. Гофф Макс К. Сетевые распределенные вычисления: достижения и проблемы - М.: „Кудиц-образ”, 2006. – 320 с.
23. Гриценко В.И., Котиков Е.А., Урсатьев А.А. и др. Модель распределенной информационной системы широкого применения // УСиМ.-1999.-№5- С.32.-42.
24. Гриценко В.И., Урсатьев А.А. Распределенные информационные системы. Состояния. Перспективы развития // Управляющие системы и машины. №4, 2003, - с.11-21.
25. Грицык В. В. Распараллеливание алгоритмов обработки информации в системах реального времени. – Киев: Наук. думка, 1981. –216 с.
26. Грои Д. Методы идентификации систем. - М.: Мир, 1979. - 304 с.
27. Дейнека В.С. Сергиенко И.В. Модели и методы решения задач в неординарних средах. – Киев: Наук. думка, 2001. – 606 с.
28. Дейнека В.С. Сергиенко И.В. Скопецкий В.В. Матиматичиские модели и методы расчьота задач с разрывными. – Киев: Наук. думка, 1995. – 262 с.
29. Згуровський М.З. Вступ до комп'ютерних інформаційних технологій: Навч. посіб. / М.З. Згуровський, І.І. Коваленко, В.М. Міхайленко; 2-е вид. – К.: Вид-во Європ. ун-ту, 2006. – 262 с..
30. Дейтел Х.М., Дейтел П.Дж., Чофнес Д.Р. Операционные системы. Распределенные системы, сети, безопасность - М.: БИНОМ, 2006. – 704 с.
31. Денисова О.О. Інформаційні системи і технології в юридичній діяльності. - К.: КНЕУ. - 2003.- 315с.
32. Дивак М.П. Задачі математичного моделювання статичних систем з інтервальними даними. Тернопіль: Видавництво ТНЕУ «Економічна думка», 2011. – 216с.
33. Дуж Я. Организация системы информации на предприятии. - М.: Прогресс, 1972. - 352 с.
34. Задірака В.К., Олексюк О.С. Комп’ютерна арифметика багаторозрядних чисел: Наукове видання.-Київ, 2003.-264с.
35. Задірака В.К., Олексюк О.С. Компютерна криптологія: Підручник.-Київ:2002.-504с.
36. Задірака В.К., Олексюк О.С. Методи захисту фінансової інформації: Навчальний посібник.-Тернопіль:Збруч, 2000.-460с.

37. Ивахненко А.Г., Лапа В.Г. Предсказание случайных процессов. - Киев: Наукова думка, 1971. - 416 с.
38. Ильченко М.Е. Сотовые радиосети с коммутацией пакетов: Моногр. / М.Е. Ильченко, С.Г. Бунин, А.П. Войтер. – К.: Наук. думка, 2003. – 265 с..
39. Кастельс М. Информационная эпоха: экономика, общество, культура: пер. с англ., под науч. ред Шкаратана О.И.- М.: ГУ ВШЭ, 2000.- 351 с.
40. Колмогоров А.Н. Теория передачи информации. - М.: Изд.АН СССР, 1956. - 230 с.
41. Колмогоров А. Н. Теория вероятности и математическая статистика. [Сб.] / Отв. ред. Ю. В. Прохоров; [АН СССР, Отд-ние математики]. – М.: Наука, 1986. – 534 с.
42. Кормич Б.А. Інформаційна безпека: організаційно-правові основи.- К.: Кондор, 2004.- 384с.
43. Корнеев И.К., Ксандопуло Г.Н., Машурцев В.А. Информационные технологии.: - М.: Проспект, 2007.- 854 с.
44. Коуров Л.В. Информационные системы и сети. Мн.: Издание НИУП, 1997.
45. Краус М. Сбор данных в управляющих вычислительных системах / Краус М., Кучбах Э., Вошни О.; пер. с нем. – М.: Мир, 1987. – 294 с.
46. Кузьмин И.В. Основы теории информации и кодирования / И.В.Кузьмин, В.А.Кедрус. – К.: Вища школа, 1986. – 238 с.
47. Кузьмин И.В., Кедрус В.А. Основы информации и кодирования. – К.: Вища школа, 1986.– 238с.
48. Лапа В. Г. Математические основы кибернетики. – К.: И-во “Выща школа”, 1974. – 452 с.
49. Макс Ж. Методы и техника обработки сигналов при физических измерениях: Пер. с франц. - М.: Мир, 1983. - Т.1 - 311 с.,Т.2 - 256 с.
50. Малиновский Б.М. Введение в кибернетическую технику. Параллельные структуры и методы / Малиновский Б.М., Боюн В.П. Козлов Л.Г. – К.: Наукова думка, 1989. – 272 с.
51. Малиновский Б.Н. Основы проектирования управляющих машин промышленного назначения / Малиновский Б.Н. – М.: Машиностроение, 1969. – 344 с.
52. Малиновский Б.Н., Боюн В.П., Козлов Л.Г. Введение в кибернетическую технику. Параллельные структуры и методы. – К.: Наук. думка, 1989. – 272с.
53. Мартин Дж. Введение в сетевые технологии. Практическое руководство по организации сетей – СПб.: Лори, 2002. – 659 с.
54. Мартин Дж. Вычислительные сети и распределенная обработка данных . – М.: Финансы и статистика, 1985. – 256 с.
55. Мартин Дж. Организация баз данных в вычислительных системах. - М.: Мир, 1980. - 662 с.
56. Мельник А.О. Архітектура комп’ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.
57. Мельник А.О. Програмовані процесори обробки сигналів.- Львів: Видавництво Національного університету «Львівська політехніка», 2000. 55с.
58. Месарович М., Такахара Я. Общая теория систем: математические основы:

Пер.с англ.– М.: Мир, 1978.– 457 с.
59. Миддлтон Д. Введение в статистическую теорию связи. – М: Советское радио, 1961. – 768 с.
60. Мирский Г. Я. Радиоэлектронные измерения. – М. Энергия, 1975. – 600 с.
61. Мирский Г. Я. Характеристики стохастической взаимосвязи и их измерение. – М: Энергоиздат, 1982. – 240 с.
62. Михалевич М.В., Сергиенко И.В. Моделирование переходной экономики:методы, модели, информацыионные технологии.-К.:Наукова думка, 2005.-671с.
63. Мінченко А.В. Правова інформатика. Концепція інформатизації: Навчальний посібник. - К.:Арістей.- 2003.- 286с.
64. Могилев А.В., Листрова Л.В. Информация и информационные процессы. Социальная информатика - М.: BHV, 2006. – 240 с.
65. Модель распределенной информационной системы широкого применения / В.И. Гриценко, Е.А. Котиков, А.А. Урсатьев и др.//УСиМ.-1999.-№5- С.32-42.
66. Модин А.А. Интегрированные системы обработки данных.- М.: Наука, 1970. - 120 с.
67. Молчанов А.А. Моделирование и проектирование сложных систем .– К.: Высш. шк., 1988.–359 с.
68. Морозов В.К., Долганов А.В. Основы теории информационных сетей. - М.: Высш.школа, 1987. - 271 с.
69. Николайчук Я.М. Теорія джерел інформації / Николайчук Я.М. - Тернопіль: ТНЕУ, 2008. – 536 с.
70. Николайчук Я.М., Возна Н.Я., Пітух І.Р. Проектування спеціалізованих комп'ютерних систем. Навчальний посібник / - Тернопіль: ТзОВ "Терно-граф". 2010. – 392с., іл.
71. Николайчук Я.М., Возна Н.Я. Пітух І.Р., Кочан В.В. Проектування спеціалізованих комп'ютерних систем. Видання друге. Навчальний посібник / Тернопіль: ТзОВ "Терно-граф". 2011. – 396с., іл.
72. Николайчук Я.М., Пітух І.Р., Возна Н.Я. Теорія моделей руху даних розподілених комп'ютерних систем. Навчальний посібник / Тернопіль: ТзОВ "Терно-граф". 2008. – 216с.
73. Оливер Б. Эффективное кодирование / Теория информации и её применение / Под ред. А. А. Харкевича. – М.: Физматгиз, 1959. – С. 159-190.
74. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб: Питер, 2002.– 544с.
75. Олифер В.Г., Олифер Н.А., Компьютерные сети. Принципы, технологии, протоколы. –СПб: Питер, 2000.– 672с.
76. Орищенко В.И. и др. Сжатие данных в системах сбора и передачи информации. - М.: Радио и связь, 1985. - 184 с.
77. Орнатский П.П. Автоматические измерения и приборы (аналоговые и цифровые) / Орнатский П. П. – [5-е изд.]. – К.: Вища шк., 1986. – 504 с.
78. Орнатский П.П. Теоретические основы информационно-измерительной техники: учебник / Орнатский П. П. – [2-е изд.]. – К.: Вища шк., 1983. – 455 с.
79. Основы моделирования сложных систем /Л.И. Дыхненко, В.Ф. Кабаненко,

И.В. Кузьмин и др.– К: Вища шк., 1981.– 246 с.
80. Основы теории вычислительных систем /С.А. Майоров и др.. – М.: Высш. шк..,1978.–408 с.
81. Основы теории информации и кодирования / И.В.Кузьмин, В.А.Кедрус. - Киев: Вища школа, 1986. - 238 с.
82. Основы цифровой обработки сигналов / А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б. Соловьева. – [2-е изд.]. – СПб.: БХВ-Петербург, 2005. – 768с.
83. Палагин А.В. Микропроцессорные вычислительные системы обработки информации: проектирования и отладка / А.В. Палагин, Е.Л. Денисенко, Р.И. Белицкий, В.И. Вигалов. – К.: Наукова думка, 1993. – 352с.
84. Палагин А.В. Системная интеграция средств компьютерной техники: Моногр. / А.В. Палагин, Ю.С. Яковлєв – Вінниця: УНІВЕРСУМ-Вінниця, 2005. — 680 с.
85. Палагин А.В., Николайчук Я.Н. Опыт разработки микропроцессорных распределенных систем реального времени. - Киев: Знание, 1988. – 19 с.
86. Палагин А.В., Яковлев Ю.С. Системная интеграция средств компьютерной техники. Монографія. – Вінниця: УНИВЕРСУМ – Вінниця, 2005. – 680с.
87. Параллельная обработка информации. Т.3 / Под ред. З.В.Грицыка. - Киев: Наукова думка, 1986. - 278 с.
88. Пасічник В.В., Резніченко В.А. Організація баз даних та знань.–К.: Видавнича група BHV, 2006.–384с.
89. Пестриков В., Маслобоев А. Delphi на примерах. – Л.: КБП, 2005. – 496 с.
90. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
91. Пономаренко В.С. Проектування інформаційних систем.–К:“Академія”, 2002.– 488с.
92. Рабин М. Основы современной системотехники. - М.: Мир, 1975. - 528 с.
93. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. - М.: Мир, 1978. - 848 с.
94. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов. - М.: Радио и связь, 1981. - 496 с.
95. Самофалов К. Г., Луцкий Г. М. Основы теории многоуровневых конвейерных вычислительных систем. – Москва: Радио и связь, 1989. – 272 с.
96. Саридис Дж. Самоорганизующиеся стохастические системы управления. - М.: Мир, 1980. - 400 с.
97. Сергієнко І.В. Виклики часу в кібернетичному вимірі.К.: Академперіодика,- 2007.-274с.
98. Сергієнко І.В. Інформатика в Україні: становлення, розвиток, проблеми/ Відп. ред: Капітонова Ю.В., Лебедева Т.Т.; НАН України, Ін-т кібернетики ім..В.М.Глушкова.-К.:Наукова думка.1999.-354с.
99. Сергієнко І.В. Оптимальні алгоритми обчислення інтегралів від швидко осцилюючих функцій та їх застосування. Том 1 Алгоритми / І.В.Сергієнко, В.К.Задірака, О.М.Литвин, С.С.Мельникова, О.П.Нечуйвітер. – Київ: Наук.думка, 2011. – 448 с.

100. Сергієнко І.В. Оптимальні алгоритми обчислення інтегралів від швидко осцилюючих функцій та їх застосування. Том 2 Застосування / І.В.Сергієнко, В.К.Задірака, О.М.Литвин, С.С.Мельникова, О.П.Нечуйвітер. – Київ: Наук.думка, 2011. – 348 с.
101. Сергієнко І.В., Дейнека В.С., Білоус М.В. Інформаціцйна технологія НАДРА-3D. Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2011. - №7. Т1. – С.153-160.
102. Современные методы идентификации систем: Пер. с англ./ Под ред. А.П.Эйкхоффа. - М.: Мир, 1983. - 400 с.
103. Солодовников А. И. Основы теории и методы спектральной обработки информации / А. И. Солодовников, А. М. Спиваковский. – Л.: Изд-во Ленингр. ун-та, 1986. – 272 с.
104. Солодовников В.В., Семенов В.В. Спектральная теория нестационарных систем управления. - М.: Наука, 1974. - 335 с.
105. Справочник по цифровой вычислительной технике: Процессоры и память / Б.Н. Малиновский, Е.И. Брюхович, Е.Л. Денисенко и др. / Под. ред. Б.Н. Малиновского. – К.: Техника, 1979. – 366с.
106. Стеклов В.К., БеркманЛ.Н. Проектування телекомунікаційних мереж.–К.: Техніка, 2002.–792с.
107. Столлингс В . Современные компьютерные сети .– СПб Питер, 2003.– 783 с.
108. Столлингс В. Беспроводные линии связи и сети.: Пер. с англ. – М.: Издательський дом «Вильямс», 2003. – 640 с.
109. Столлингс В. Передача данных.– СПб.: Питер, 2004.– 750 с.
110. Столлингс В. Структурная организация и архитектура компьютерных систем., 5-е изд.: Пер. с англ. - М.: Издательський дом "Вильямс", 2002. - 896с.
111. Таненбаум Є. Современные компьютерные сети.–СПб.: Питер, 2003.–992с.
112. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. - СПб.: Питер, 2003. - 880с.
113. Тихонов А.Н., Арсенич В.Я. Методы решения некорректных задач. – М.: Наука, 1979. – 287с.
114. Тихонов В.И., Миронов М.А. Марковские процессы. - М. Со в.радио, 1977. - 488 с.
115. Точчи Р., Рональд Дж., Уидмер, Нил С. Цифровые системы. Теория и практика.: Пер. с англ.–М. : Издательский дом „Вильямс”, 2004.– 1024 с.
116. Тузов Г.И. Статистическая теория приема сложных сигналов. - М.: Сов.радио, 1977. - 400 с.
117. Тчауб Дж., Василковский Т., Вожняковский Х. Информация, неопределенность, сложность: Пер.с англ.-М.:Мир, 1988.-184с.
118. Финк Л.М. Теория передачи дискретных сообщений. - М.: Советское радио, 1970. - 728 с.
119. Флейшман Б.С. Основы системологии. – М.: Радио и связь, 1982.– 342 с.
120. Френкс Л. Теория сигналов / Френкс Л.; пер. с англ. Д. Е. Вакмана. – М: Сов. радио, 1974. – 344 с.
121. Фритч Б. Применение микропроцессоров в системах управления: Пер. с нем. - М.: Мир, 1984. - 464 с.

122. Харкевич А.А. Борьба с помехами. - М.: Физматгиз, 1963. - 136 с.
123. Харкевич А.А. Очерки общей теории связи. - М.: Гос-техиздат, 1955. - 268 с.
124. Харкевич А.А. Спектры и анализ. – М.: ГИТТЛ, 1957.– 280 с.
125. Харкевич А.А. Теория информации. Опознавание образов, т. Ш. - М.: Наука, 1973. - 524 с.
126. Хармут Х.Ф. Передача информации ортогональными функциями. - М.: Связь, 1975. - 275 с.
127. Хартли Р. Передача информации / Теория информации и ее приложения./ Ред. А.А. Харкевич. - М.: Физматгиз, 1959, с.5-36.
128. Цветков Э.И. Основы теории статистических измерений. Л.: Энергоатомиздат, 1986. - 256 с.
129. Цикритзис Д., Лоховски Ф. Модели данных.- М.: Финансы и статистика, 1985.-343 с.
130. Цифровая связь. Д. Прокис; Пер. с англ., ред Д. Д. Кловский. – М.: Радио и связь, 2000. – 797с.
131. Шахнович И. Современные технологии беспроводной связи. М., Техносфера, 2006 – 166с.
132. Шеннон К. Работы по теории информации и кибернетике. - М.: Изд-во иностр. лит, 1963. – 438 с.
133. Шеннон К. Статистическая теория передачи электрических сигналов / Теория передачи электрических сигналов при наличии помех. - М.: Изд-во иностр.лит-ры, 1953, С.7-85.
134. Шеннон К.Э. Теория связи в секретных системах. В работах по теории информации и кибернетики.-М.: И.Л., 1963.-830с.
135. Э.Квейд Анализ сложных систем / Пер. с англ. под ред. И.И.Ануреева, И.М.Верещагина. – М.: Советское радио, 1969, 520с.
136. Якубайтис Э.А. Информационно-вычислительные сети. – М.: Финансы и статистика, 1984. - 232 с.
137. Якубайтис Э.А. Локальные информационно-вычислительные сети. - Рига: Зинайте, 1985. - 284 с.

Теоретичні основи та застосування кодів поля Галуа.

1. Алексеев А.И. и др. Теория и применение псевдослучайных сигналов. - М.: Наука, 1969. - 366 с.
2. Алишов Н.И. RH-оптимізація в мережахкомутації макутів. Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2011. - №7. Т1. – С. 16-20.
3. Аллен Дж. Архитектура процессоров для цифровой обработки сигналов // ТИИЭР, 1985, т.73, № 5, с.3-37.
4. Амиантов И.Н. Избранные вопросы статистической теории связи. - М.: Сов.радио, 197I. - 416 с.
5. Анисимов А.В. Конвейерное вычисление элементарных функций по методу цифра за цифрой// Известия ЛЭТИ: сб.науч.тр.: -Л. – 1991. – Вып.438. – С. 20-23.

6. Анисимов Б. В., Курганов В. Д., Злобин В. Н., Распознавание и цифровая обработка изображений. – М.: Высш. Школа, 1983. – 295 с.
7. Антонов А.П. Язык описания цифровых устройств-Altera-HDL, М: "Радиософт", 2001, 224с.
8. Апнезет К., Дзун Д., Кьесбю С., Шайбль Г., Циммерман В. Техника беспроводной связи. Беспроводные датчики ближней локации // АББ Ревю. – 2002. – №4. – С. 42-47.
9. Бабак В. П. Детерміновані сигнали і спектри: [навч. посіб. для студ. вищ. навч. закл.] / В. П. Бабак, А. Я. Білецький; пер. з рос. – К.: Техніка, 2003. – 455 с.
10. Бабич Н. П. Компьютерная схемотехника. Методы построения и проектирования: учебное пособие / Н. П. Бабич, И. А. Жуков. – К.: МК–Пресс, 2004. – 576 с.
11. Балакришнан А.В. ред. Статистическая теория связи и ее приложения. - М.: Мир, 1967. - 250 с.
12. Балашов Е.П. и др. Высокопроизводительные специализированные процессоры для вычисления элементарных функций // Электронное моделирование. - 1983. - № 4 - С. 61-65.
13. Балашов Е.П., Пузанков Д.В. Проектирование информационно-управляющих систем. - М.: Радио и связь, 1987. - 256 с.
14. Банки данных в автоматизированных системах обработки данных. - Киев: ИК АН УССР, 1981. - 176 с.
15. Баня Е.Н. Компьютерные сети – К.: Корнійчук, 2009. - 264 с.
16. Барановская Т.П., Лойко В.И., Семенов М.И. Архитектура компьютерных систем и сетей: Учеб. Пособие. –М.: Финансы и статистика, - 2003,- 256с.
17. Бат М. Спектральный анализ в геофизике / пер. с англ. – М.: Недра, 1980. – 535 с.
18. Бебих Н. В., Денисов А. И. Взаимная спектрально-корреляционная обработка сигналов в различных ортогональных базисах // Изв. вузов. Сер. Радиоэлектроника. 1983. Т. 26, № 3. С. 54-60.
19. Белецкий Б.А., Вагис А.А., Васильев С.В., Гупал А.М. Процедуры распознавания вторичной структуры белков / Проблемы управления и информатики № 4, 2007
20. Белецкий Б.А., Васильев С.В., Гупал А.М. Предсказание вторичной структуры белков на основе байесовских процедур распознавания / Проблемы управления и информатики - №1, 2007
21. Белецкий Б.А., Васильев С.В., Гупал А.М., Сергиенко И.В. Предсказание вторичной структуры белков на основе байесовских процедур распознавания на цепях Маркова / Кибернетика и системный анализ, № 2, 2007
22. Белецкий Б.А., Гупал А.М. Статистический анализ геномов бактерий. Комплементарность оснований / Проблемы управления и информатики. - №6, 2005
23. Белецкий Б.А.,Быць А.В., Гупал А.М., Ржепецкий С.С., Рязанов В.В, Сергиенко И.В. Методы предсказания пространственной структуры белков / Кибернетика и системный анализ, 2010, №1

24. Беліма А. С., Боличевцев О.Д., Гребень Й.І. Теоретичні основи централізованого контролю технологічними процесами. – К.: Вища шк., 1973. – 242 с.
25. Белоглазова О.В., Лабунец В.Г. Теория и применение преобразований Гаусса-Рэйдера. - Изд-во АН СССР. Техн.кибернетика, 1981, № 2, с.193-200.
26. Бендат Дж., Пирсол А. Применение корреляционного и спектрального анализа / Пер. с англ. - М.: Мир, 1983. - 312 с.
27. Бердышев Е. Технология ММХ. Новые возможности процессоров P5 и P6.- М.:ДИАЛОГ-МИФИ, 1998.- 234 с.
28. Бертсекас Д., Галлагер Р. Сети передачи данных: Пер. с англ. - М.: Мир, 1989. - 544 с.
29. Блейхут Р. Теория и практика кодов контролирующих ошибки. – М.: Мир, 1986. – 576с.
30. Бойченко Е.В. Методы схемотехнического проектирования распределительных информационно-вычислительных сетей микропроцессорных систем. - М.: Энергоатомиздат, 1988. - 128 с.
31. Борисенко А. А. Системы счисления и ЭВМ / А. А. Борисенко // Вестник СумГУ. – 1996. – № 2 (6). – С. 72 – 75.
32. Боумгарт В.Ф., Зибинь Д.К., Трайнин С.Б. Базовая ЛВС для широкого применения / Материалы III Всесоюзной конференции "Локсеть-86". - Рига: ИЭВТ, 1988, Т.1, с.34-38.
33. Брызгалов А.П. Базовая корреляционная функция сверхширокополосных сигналов большой длительности. Труды ГосНИИАС, серия “Авионика”, 2000, вып 3, - с.7-15.
34. Брюхович Е.И. Экономическая стратегия разработки вычислительных систем: место и роль счислений // Управляющие системы и машины. - № 2, 1990,- с.3-18.
35. Брянцев И.Н Data Mining. Теория и практика - М.: БДЦ-Пресс, 2006. – 598 с.
36. Бунин С. Г., Войтер А.П. Вычислительные сети с пакетной радиосвязью. – Киев: Техніка, 1989. – 223с.
37. Бунин С.Г. Пакетная радиосвязь передачи данных на основе сигналов с расширенной базой / Высокопроизводительные преобразователи формы информации и средства передачи данных. - Киев: ИК АНУССР, 1984, с.46-51.
38. Бунин С.Г. Применение сложных сигналов в информационно-вычислительных сетях с пакетной радиосвязью / Технологические средства обработки для высокопроизводительных ЭВМ и систем. - Киев: ИК АН УССР, 1984, с.46-51.
39. Бунин С.Г., Войтер А.П. Эффективность применения шумоподобных сигналов в сетях передачи данных / Средства передачи, преобразования и обработки информации для высокопроизводительных систем и сетей. - Киев: ИК АН УССР, 1985. - 34-38 с.
40. Бунин С.Г., Войтер А.П., Пилипчак С.И. Протоколы множественного доступа для больших локальных сетей / Локальные вычислительные сети. - Рига: ИЭВТ, 1988, том I, с.59-62.
41. Буров Є. Комп’ютерні мережі. – Львів: БаК, 1999. – 468 с.

42. Бусленко Е.А. и др. Лекции по теории сложных систем.– М.: Сов. радио, 1973.–283 с.
43. Быць А.В., Гупал А.М., Яремчук Т.Г. Статистический анализ генома нематоды C.elegans. Закономерности записи комплементарных азотистых оснований / Доклады НАНУ, 2006, № 2
44. Вагис А.А., Гупал А.М. Комплементарность оснований в хромосомах ДНК / Проблемы управления и информатики. - №5, 2005
45. Вагис А.А., Гупал А.М. Математика и живая природа. Удивительный мир ДНК./ Проблемы управления и информатики / №1-2, 2006
46. Вагис А.А., Гупал А.М., Принципы организации живой природы / Сб. Компьютерная математика, ИК НАНУ, 2005, №1
47. Вагис А.А., Гупал А.М., Сергиенко И.В., Соотношения комплементарности в записи оснований по одной нити в хромосомах ДНК / Проблемы управления и інформатики. - №4, 2005
48. Варакин Л.Е. Системы связи и шумоподобными сигналами. - М.: Радио и связь, 1985. 384 с.
49. Варакин Л.Е. Системы связи с шумоподобными сигналами. – М.: Радио и связь, 1985. –384 с.
50. Васильев С.В., Гупал А.М. Исследование особенностей записи оснований в ДНК на кластерном компьютере / Сб. Компьютерные средства, сети и системы ИК НАНУ, 2006, № 5
51. Вебер Д.II. Экономический аспект проблемы сжатия данных . / В кн.: Достижения в области телеметрии. - М.: Мир, 1970. - 214 с.
52. Венда В.Ф. Инженерная психология и синтез систем отображения информации. - М.: Машиностроение, 1975. - 396 с.
53. Веников В.А Теория подобия и моделирования – М.: Высш. шк., 1976.– 384с.
54. Вербицький О.В. Вступ до криптології.- Львів: Вид-во наук.тех. л-ри, 1998.- 248с.
55. Вербовецкий А.А. Оптическая голография в цифровых компьютерных технологиях, - М., Алекс-Верб, 2004, - 149с.
56. Возна Н.Я Інформаційні технології формування техніко-економічних даних для об’єктів управління різних класів стаціонарності Міжнародний науково-технічний журнал “Оптико-електронні інформаційно-енергетичні технології”. — 2011. — №2(22). — С. 74-78.
57. Возна Н.Я Теорія та методи побудови моделей руху даних у розподілених КС Вісник національного університету "Львівська політехніка" "Комп’ютерні системи та мережі". – 2010. - №688. – С.60-64.
58. Возна Н.Я. Формалізація моделей руху даних розподілених комп’ютерних систем та оцінювання їх структурної складності. Науковий журнал "Вісник Тернопільського національного технічного університету ім..І..Пулюя" – 2011. - №1. Т.16. – С.167-179.
59. Возна Н.Я., Николайчук Я.М. Метод формування та організації руху даних у розподілених комп’ютерних системах на основі багаторівневих матричних моделей. Праці міжнародного симпозіуму: "Питання оптимізації обчислень

(ПОО-XXXV)". – Київ: Інститут кібернетики ім. В.М.Глушкова НАН України, 2009. – Т.1. – С.120-125.
60. Возна Н.Я., Николайчук Я.М. Теорія моделей джерел інформації та формування ідентифіковано-структуризованих даних комп'ютеризованих систем. Научно-теоретический журнал "Искусственный интеллект". ІПШІ МОН і НАН України "Наука і освіта". – 2008. - №5. – С.342-349.
61. Волошинов С.Д., Николайчук Я.Н., Петришин Л.Б. Структура и организация оперативной памяти с параллельным многопроцессорным доступом / Распараллеливание обработки информации, кн.1. - Львов, ФМИ, 1989, с.173-174.
62. Галушкин А. И., Золотов Ю. Я., Шикулов Ю. А. Оперативная обработка экспериментальной информации. - М. : Энергия, 1972. -360 с.
63. Гитис Э. И. Аналого-цифровые преобразователи: учеб. пособие для вузов / Э. И. Гитис, Е. А. Пискулов. – М.: Энергоиздат, 1981. – 360 с.
64. Гоноровский И. С. Радиотехнические цепи и сигналы: учеб. для вузов / Гоноровский И. С. – [4-е изд.]. – М.: Радио и связь, 1986. – 512 с.
65. Горелов Г.В. Нерегулярная дискретизация сигналов.- М.: Радио и связь, 1982. - 256 с.
66. Грибанов Ю.И., Веселова Г.П., Андреев В.Н. Автоматические цифровые корреляторы. –М.: Энергия, 1971.–240с.
67. Гридина Н.Я, Гупал А.М., Палагин А.В., Тарасов А.Л. Автоматизированная система анализа показателей скорости оседания эритроцитов при глиомах головного мозга/ Проблемы управления и информатики. – 2009. –№ 3. – С.136-143.
68. Гридина Н.Я, Гупал А.М., Сергиенко И.В., Тарасов А.Л. Байесовская процедура распознавания глиом головного мозга / Проблемы управления и информатики. – 2009. –№ 5. – С.150-154.
69. Гридина Н.Я, Гупал А.М., Тарасов А.Л. Экспертная система анализа прогрессий глиом головного мозга/ Компьютерная математика. – 2007. – № 2. – С. 132–139.
70. Гук М. Апаратные средства IBM PC. Энциклопедия .– СПб : Питер, 2001. – 816с.
71. Гук М. Аппаратные интерфейсы ПК: Энциклопедия .- СПБ.: Питер, 2003 .- 528 с.
72. Гук. М. Аппаратные средства локальных сетей. Энциклопедия – СПб: Питер, 2000.-576с.
73. Гулевич Д.С. Сети связи следующего поколения.: - М.: БИНОМ, 2007. – 673с.
74. Гуменюк Р.М., Іщеряков С.М. Аналіз методу подвійного згортання із послідовним використанням різних статистичних функцій // Вісник Вінницького політехнічного інституту.-Вінниця : ВПІ.,–2003.- №6 с.75-79.
75. Гупал А.М., Журбенко А.Н, Пашко С.В. Перспективы исследования геномов / Сб. Компьютерная математика, ИК НАНУ, 2005, №2
76. Гупал А.М., Сергиенко И.В. Оптимальные процедуры распознавания / Наукова думка, 2008

77. Гупал А.М., Сергиенко И.В. Принципы построения процедур индуктивного вывода // Кибернетика и системный анализ, №4, 2006
78. Гупал А.М., Сергиенко И.В. Соотношения комплементарности в записи оснований по одной нити в ДНК/ Цитилогия и генетика. - №6, 2005
79. Гуревич М.С. Спектры радиосигналов. - М.: Связьиз- дат, 1963. - 312 с.
80. Дадаев Ю.Г. Теория арифметических кодов.-М.:Радио и связь, 1981.-270с.
81. Данильченко Л.С. О некоторых эффективных алгоритмах вычисления остатка и возведения в степень многорозрядных чисел//Кибернетика и системный анализ, №3, 1996.-С.145-151.
82. Девиссон Д. Скорость создания сообщений. Теория и применение / Сб.: Обработка изображений при помощи цифровых вычислительных машин. - М.: Мир, 1973, с.87-97.
83. Демихов В.И., Леов А.И. Контрольно-измерительные приборы при бурении скважин. - М.: Недра, 1980. - 240 с.
84. Денисов В.А. Выбор параметров эталонного описания стилизованных знаков / Автоматизация ввода письменных знаков в ЦВМ. - Каунас, КПИ, 1984, с.112-115.
85. Деннинг В., Эссиг Г., Маас С. Диалоговая система "Человек - ЭВМ". Адаптация и требования пользователя. - М.: Мир, 1984. - 112 с.
86. Джалиашвили З.О., Ожиганов А.А. Кодовые шкалы в оптоэлектронных преобразователях информации / Функциональная оптоэлектроника в вычислительной технике и устройствах управления. - Тбилиси, 1986. - 168 с.
87. Дженкинс Г., Ватте Д. Спектральный анализ и его приложения. - М.: Мир, 1971, вып.1. - 316 е.; вып.2. - 228 с.
88. Джонсон Дж., Говард В. Высокоскоростная передача цифровых данных.: Пер. с англ.. – М.: Издательский дом «Вильямс», 2005. – 1024 с.
89. Дивак М. П., Пітух І.Р., Шкляренко Н.П., Франко Ю.П.. Використання властивостей інтервальних похибок при моделюванні технологічних процесів// Вимірювальна та обчислювальна техніка в технологічних процесах: Збірник наукових праць.–Хмельницький: ТУП, 2000.– С. 272.
90. Дивак М.П. Властивості інтервальних моделей при інтервальній формі їх параметрів // Сб. науч. тр. международного науч.–учеб. центра информ. технологий и систем, науч. совет НАН Украины по пробл. „Кибернетика”. Моделирование и управление состоянием эколого–экономических систем региона.– К.–2001.–С.58–63.
91. Диксон Р.К. Широкополосные системы: Пер. с англ. / Под ред. В.И.Журавлева. - М.: Связь, 1979. - 288 с.
92. Диффи У., Хеллмэн М.Э Защищенность и имитостойкость: Введение в криптографию//ТИИЕР, т.67, №3, 1979.-С.71-109.
93. Домрачев В.Г. и др. Схемотехника цифровых преобразователей перемещений. - М.: Энергоатомиздат, 1987. - 392 с.
94. Дудыкевич В.Б. Специализированные периферийные процессоры для первичной обработки информации, представленной число-импульсным кодом / Диагностика и коррекция погрешностей преобразователей технологической информации, - Киев: КГИЛП, 1989, с.108-109.

95. Дунець Р.Б. Аналіз та синтез топологій комп'ютерних видавничо–поліграфічних систем. – Львів: НВФ „Українські технології", 2003. – 192с.
96. Дэвис Д., Барвер А., Прайс У. и др. Вычислительные сети и сетевые протоколы.- М.: Мир, 1982.- 196 с.
97. Елисеева И.И., Рукавишников В.О. Логика прикладного статистического анализа. - М.: Наука, 1982. - 126 с.
98. Жаровский С.Н. Моноканальная терминальная сеть ОДА-20М // УСиМ, .№6, 1983, С. 34-39.
99. Жаровский С.Н. О построении локально-региональной сети передачи данных на базе выделенных телефонных линий / УСиМ, 1986,№ 4, с.73-90.
100. Жаровский С.Н. Сетевая организация распределенных систем управлення.- К.: И-во "Знание", 1990.- 23 с.
101. Железов И.Г. Сложные технические системы (оценка характеристик). – М .: Высш.шк., 1984.– 119 с.
102. Жовинский В.Н., Арховский В.Ф. Корреляционные устройства. - М.: Энергия, 1974. - 248 с.
103. Жуган Л.И., Волошинов С.Д., Николайчук Я.Н. Сети распределенного управления технологическими объектами на базе микроЭВМ и ПЭВМ. - К.: Знание, 1989. - 20 с.
104. Жуган Л.И., Николайчук Я.Н. Логико-статистической анализ информационных состояний сложных обьектов управления / Логическое управление с использованием ЭВМ. Тезисы докладов XII Всесоюзного симпозиума. - Москва-Симферополь, 1989, с.327- 329.
105. Журавський В.С. Україна на шляху до інформаційного суспільства .– К: ІВЦ „Видавництво „Політехніка", 2004.– 484 с.
106. Жураковский Ю.П., Волошин В.И. Многочастотные системы передачи дискретных сигналов. - Киев: Техшка, 1981. - 120 с.
107. Жураковский Ю.П., Назаров В.Д. Каналы связи. - Киев Вища школа, 1985. - 232 с.
108. Жураковський Ю.П. Теорія інформації та кодування: підручник / Ю. П. Жураковський, В. П. Полторак. – К.: Вища шк., 2001. – 255 с.
109. Задирака В.К. Цифровая обработка сигналов / В.К. Задирака, С. С. Мельникова. – К.: Наук. думка, 1993. – 294 с.
110. Заставний О.М. Аналіз системних характеристик спецпроцесорів формування вихідних даних аналого-цифрових кодерів // Вісник Технологічного університету Поділля, Хмельницький, 2005, №4, ч.1, Т2. С. 223-226.
111. Заставний О.М. Синтез та проектування аналого-цифрового кодера автономного сенсора з вихідними двомірними шумоподібними сигналами // Збірник наукових праць інституту проблем моделювання в енергетиці ім. Г.Є. Пухова.- Київ, -2006, – С.34 – 42.
112. Заставний О.М. Системні параметри автономних сенсорів з глибоким розпаралеленням інформаційних потоків // Вісник Технологічного університету Поділля, -Хмельницький, -2003, -№3, -Т1, -С.128 – 131.
113. Заставний О.М., Николайчук Я.М. Методологія побудови автономних

сенсорів для розподілених комп'ютерних мереж // Вісник Технологічного університету Поділля, Хмельницький, 2002, №3, Т1.ст. 142-146.
114. Зевелев С.Я., Крикун З.Н., Николайчук Я.Н. Выбор оптимальных параметров кодирования методом вычетов // Автоматизация и телемеханизация нефтяной промышленности. - М.; №2, 1975, С. 3-5.
115. Зевелев С.Я., Николайчук Я.Н. Синтез структуры цифрового модема с трехчастотной манипуляцией // Автоматизация и телемеханизация нефтяной промышленности, № 4, 1977, С- 10-13.
116. Зелинский Д.И., Лучук А.М., Паук С.М. Приемники дискретных многопозиционных сигналов. - Киев: Наукова думка, 1976. - 240 с.
117. Зюко А.Г. и др. Помехоустойчивость и эффективность систем передачи информации. - М.: Радио и связь, 1985. -272 с
118. Зяблов В.В., Шавгулидзе С.А. Обобщенные каскадные помехоустойчивые конструкции на базе сверточных кодов. – М.: Москва, 1991. – 207 с.
119. Ибрагимов В.А., Крикун З.Н., Николайчук Я.Н. Кодирование информации методом вычетов. - // Автоматизация и телемеханизация нефтяной промышленности. № 1, 1974, С. 26-28.
120. Ионин Д.А., Яковлев Е.И. Современныё методы диагностики магистральных газопроводов. - Л.: Недра, 1987. - 232 с.
121. Іщеряков С.М., Каюк Т.П. Взаємокореляційні властивості ансамблів багаторівневих М-послідовностей // Вісник Житомирського інженерно-технологічного інституту. – Житомир : ЖІТІ – 2002 –с.83-87.
122. Іщеряков С.М., Полянчич А.Я. Структурні властивості ключів багаторівневих М-послідовностей // Вісник Технологічного університету Поділля, Хмельницький, 2005, №4 ч.1, Т2. С.65 – 68.
123. Іщеряков С.М., Федорович Ю.С. Комп'ютерне моделювання взаємокореляційних методів приймання фазоманіпульованих гармонійних сигналів // Вісник Житомирського інженерно-технологічного інституту. – Житомир : ЖІТІ – 2002 – С.28-32.
124. Како Н., Яманэ Я. Датчики и микро-ЭВМ. - Л.: Энерго- атомиздат, 1986. - 120 с.
125. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука, 1983. - 424 с.
126. Каляев А.В. Многопроцессорные системы с перестраиваемой архитектурой. - М.: Радио и связь, 1984. - 240 с.
127. Капеллин В. и др. Цифровые фильтры и их применение.-М.: Энергоатомиздат, 1983. - 360 с.
128. Катренко А.В. Системний аналіз об'єктів та процесів комп'ютеризації: Навч. посібник. –Львів: «Новий світ – 2000», 2001. – 424 с.
129. Кей С. М., Марпл С. Л. Современные методы спектрального анализа. // ТИИЭР. Т. 69 – № 11. –1981. – С. 3–51.
130. Кларк Дж, Кодирование с исправлением ошибок в системах цифровой связи: Пер. с анг.- Н.: Радио и связь, 1987. - 468 с.
131. Клопов В.А., Мотуз О.В. Основы компьютерной стеганографии// Конфидент,4, 97.С.43-48.

132. Коваленко А.М. САПР: Методология и формализированные методы Л., 1988.-120 с.
133. Козловский Е.А., Питерский В.М., Комаров М.А.,Кибернетика в бурении. - М.: Недра, 1982. - 298 с.
134. Кондалев А.И. Высокопроизводительные преобразователи формы информации / Кондалев А.И. и др. – К.: Наук. думка, 1987. – 280 с.
135. Кондалев А.И. Преобразователи формы информации для малых ЭВМ / Кондалев А.И. и др. – К.: Наук. думка, 1982. – 312 с.
136. Кондалев А.И., Еремеев М.С. Интеллектуальные терминалы - новый этап в развитии преобразователей формы информации / Проблемы создания преобразователей формы информации. - К.: Наук. думка, 1980, ч. 1, с. 84-86.
137. Конюховский П.В., Колесов Д.Н. Экономическая информатика.- СПб,: 2001.-560 с.
138. Кормен Т. Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. - СПб.: Вильямс, 2005.-1296 с.
139. Корнеев В.В. Паралельные вычислительные системы. - М.: Нолидж, 1999. - 320 с.
140. Коуги П.М. Архитектура конвейерных ЭВМ: Пер. с англ. - М.: Радио и связь, 1985. - 360 с.
141. Коутс Л., Олейник И. Интерфейс „человек - компьютер”.- М.: Мир, 1993.- 400 с.
142. Криксунов В. Г. Спектральный анализ электрических сигналов. – К.: Техника, 1971. – 195 с.
143. Крикун З.Н., Зевелев С.Я., Николайчук Я.Н. Расчет параметров цифрового трехчастотного манипулятора для систем передачи данных / Элементы технических средств АСУ нефтяной промышленности. - Киев: Техшка, 1977, С. 39- 44.
144. Крикун З.Н., Николайчук Я.Н., Божнев В.П. Представление измерительной информации в нормализованной системе счисления остаточных классов / Известия ВУЗов "Нефть и газ", № 6, 1976, С. 81-86.
145. Крикун З.Н., Николайчук Я.Н., Лабий О.М. Комплекс технических средств контроля и управления процессом бурения. Материалы IX Всесоюзного научно-технического совещания по АСУ ТП, 1980, С. 14.
146. Крикун З.Н., Николайчук Я.Н., Ширмовский Г.Я. Выбор алгоритма и схемы умножения в устройствах; цифрового уплотнения измерительной информации / Разведка и разработка нефтяных и газовых месторождений. - Львов, № 12, 1975, С. 142-148.
147. Круцкевич Н.Д. Перспективи розвитку комп’ютерних мереж з реконфігурацією архітектури на базі пам’яті колективного доступу // Вісник «Радіоелектроніка та телекомунікації» Національного університету ”Львівська політехніка”, Львів, 2004, №508, С. 240 – 245.
148. Круцкевич Н.Д. Принципи побудови дешифраторів кодів Галуа пам’яті колективного доступу // Вісник Технологічного університету Поділля, - Хмельницький, -2004,- №2, -Ч.1, Т2, -С. 113 – 116.
149. Кузьмин С.З. Основы проектирования систем цифровой обработки

радиолокационной информации.–М.: Радио и связь, 1986.– 352 с.
150. Курочкин С.С. Многоканальные счетные системы и коррелометры. - М.: Энергия, 1972. - 344 с.
151. Кухарев Г.А., Тропченко А.Ю., Шмерко В.П. Систолические процессоры для обработки сигналов. - Минск: Беларусь, 1988. - 128 с.
152. Лазарович І.М. Николайчук Я.М. Метод рандомізації та цифрової обробки інформаційних потоків в системах автоматизації виробничих процесів. // Вісник технологічного університету Поділля, №3 – Хмельницький, 2002. Т.2, С.91-94.
153. Лазарович І.М., Николайчук Я.М. Теорія і методи рандомізації цифрових потоків в телекомунікаційних системах. Вісник національного універ-ситету "Львівська політех-ніка" Радіоелектроніка та телекомунікації.-2002.-№443.-С.234-240.
154. Ланге Ф.Г. Корреляционная электроника. - Л.: Судпром гиз, 1963. - 448 с.
155. Левин Б.Р. Теоретические основы статистической радио техники. - М.: Сов.радио, 1966, Т.1. - 326 с.
156. 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.
157. 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.
158. Левицький А. О. Аналіз методики обчислення витрати енергоносіїв при використанні вертикальної інформаційної технології. // Методи та прилади контролю якості. – Івано-Франківськ, 1998. – №2. – С. 63–65.
159. Литвин А.И., Май А.И., Писаренко Л.А. Организация векторных вычислений спектральных коэффициентов преобразования Хаара // Тезисы докладов междунар. конф. по вычислительной математике (МКВМ-2002).-Новосибирск.- 2002. – С.46-58.
160. Логунова О. С., Ячиков И. М., Ильина Е. А. Человеко-машинное взаимодействие: теория и практика.- Ростов-на-Дону.: ФЕНИКС, 2006.- 612с.
161. Локазюк В.М. Проблеми та методологія контролю і діагностування сучасних мікропроцесорних пристроїв та систем // Вимірювальна та обчислювальна техніка в технологічних процесах.– 2000.- № 2.–с.10–17.
162. Локазюк В.М. Контроль і діагностування обчислювальних пристроїв та систем : Навч. посібник для вузів .– Хмельницький : ТУП, 2001.–242 с.
163. Локазюк В.М., Поморова О.В., Домінов А.О. Інтелектуальне діагностування мікропроцесорних пристроїв та систем: Навч. посібник для вузів. Хмельницьк 2001.–286с.
164. Лучук А.М. Генерирование и разделение частотных сигналов. - Киев: Техніка, 1966. - 180 с.
165. Лучук А.М. Устройства передачи дискретной информации. - К.: Техніка, 1976. - 260 с.

166. Лучук А.М., Бунин С.Г., Бучкин А.М. Сеть передачи дискретной информации на основе радиоканала с множественным доступом / Проблемы преобразования и передачи информации. - Киев: ИН АН УССР, 1980, С. 63 - 67.
167. Лучук А.М., Жаровский С.Н. Передача данных по многопунктовому каналу связи в низовых звеньях сложных систем // Механизация и автоматизация управления, 1978, № 3, с. 73 - 75.
168. Майника Э. Алгоритмы оптимизации на сетях и графах.–М.: Мир, 1981.– 323с.
169. Майоров С.А., Новиков Г. И. Принципы организации цифровых машин. Л., "Машиностроение", 1974, 432с.
170. Макаров С.Б., Цикин И.А. Передача дискретных сообщений по радиоканалам с ограниченной полосой пропускания. - М.: Радио и связь, 1988. - 304 с.
171. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов исправляющих ошибки.: Пер. с англ. – М.: Связь, 1979. –744 с.
172. Малые локальные сети микроконтроллеров и микро-ЭВМ // Обзоры по электронной технике, серия 3, микроэлектроника. М:. ЦНИИ, 1987. – 64 с.
173. Мановцев А.П. Введение в цифровую радиотелеметрию. - М.: Энергия, 1967. - 164 с.
174. Марков С. Цифровые сигнальные процессоры. Книга 1. -М.: фирма МИКРОАРТ, 1996.- 144 с.
175. Мартин Дж. Планирование развития автоматизированных систем – М.: Финансы и статистика, 1984. – 196с.
176. Мартин Дж. Системный анализ передачи данных.-- М.: Мир, 1975 - 256с.
177. Марущак А.І. Інформаційне право: Доступ до інформації. – К.: КНТ, 2007. – 532с.
178. Марущак А.І. Правові основи захисту інформації з обмеженим доступом, Курс лекцій.-К.КНТ, 2007.-208с.
179. Машбиц Л.М. Цифровая обработка сигналов в радиотелеграфной связи. - М.: Связь, 1974. - 192 с.
180. Мелик-Шахназаров А.М. Информационно-вычислительные системы и их применение в нефтяной промышленности // Изв. ВУЗов: Нефть и газ, 1970. - № 4, С. 108-112.
181. Мелик-Шахназаров А.М., Браго Е.Н., Савин. В.В. Методы и устройства преобразования, сжатия и отображения информации для обьектов нефтяной промышленности. / Материалы Всесоюзной конференции "ИИС-73". - Ивано-Франковск, 1974. - С. 5-9.
182. Мельничук С.І. Малоенергетичні методи завадостійкого обміну даними в безпровідних комп'ютерних мережах автоматизованих систем. // BISTRO/ 96/052 Матеріали 2-ї Міжнародної науково-практичної конференції "Управління енерговикористанням". - Львів. - 1997. - С.47 - 50.
183. Мельничук С.І. Основи автоматизованого проектування елементів та засобів обчислювальної техніки : Навчально–методичний посібник.– Івано–

Франківськ : Видавництво ІМЕ, 2004.–180 с.
184. Методы и средства обработки диагностической информации в реальном времени / В.А.Гуляев, В.М.Чаплыга, И.В.Кедровский. - Киев: Наукова думка, 1986. - 224 с.
185. Муттер В.М. Основы помехоустойчивой телепередачи информации. – Л.: Энергоатомиздат. Ленингр. отд-ние, 1990. – 268 с.
186. Назаров М.В., Прохоров Ю.Н. Методы цифровой обработки и передачи речевых сигналов. М.: Радио и связь, 1985. - 176 с.
187. Нейфах А.Э. Сверточные коды для передачи дискретной информации. – М.: Наука, 1979. – 222 с.
188. Николаев В.И., Брук В. М. Системотехника: методы и приложения .–Л.: Машиностроение, 1985.–684 с.
189. Николайчук Л.М. Особливості побудови характеристик продукційних моделей подачі юридичних знань. // Науковий журнал. -Вісник Хмельницького національного університет-ту. - Хмельницький, 2006, № 5 (85).- С.113- 115.
190. Николайчук Л.М. Формалізація норм та часових характеристик юридичних законів на основі логіко-статистичних інформаційних моделей // Збірник наукових праць.-Національна академія наук України Інститут проблем моделювання в енергетиці ім. Г.Є.Пухова. - Випуск 38.- Київ. – 2006 С.44 -50.
191. Николайчук Л.М., Безштанько О.П. Формалізація та інформатизація юридичних знань на основі продукційних моделей та графів взаємодії суб'єктів юриспруденції. // Научно-теоретический журнал "Искусственный интеллект". ІПШІ МОН і НАН України "Наука і освіта". – 2008. - №4. – С.395-402.
192. Николайчук Л.М., Возна Н.Я.. Реалізація цифрового підпису в телекомунікаційних системах та його правові аспекти. //Науковий жунал.- Вісник Технологічного університету Поділля.-№ 3 Том. 1(51). - Хмельницький, 2003. – С. 125-128.
193. Николайчук Л.М., Чегодар О.М. Проблеми створення інформаційних систем юридичних знань та оцінки ентропії юридичної інформації. // Науккові вісті.- Інститут менеджменту та економіки «Галицька академія».- № 2 (10).- 2006.- С. 154-161.
194. Николайчук Я. Н. Разработка теории и комплекса технических средств формирования, передачи и обработки цифровых сообщений в низовых вычислительных сетях автоматизированных систем: диссертация на соискание ученой степени доктора технических наук / Николайчук Ярослав Николаевич. – Ивано-Франковск: ИФИНГ, 1991. – 573 с.
195. Николайчук Я. М., Возна Н.Я. Пристрій для введення алфавітно-цифрових даних. Патент на корисну модель № 25291.– 2007р.
196. Николайчук Я., Король Р. Вертикальна інформаційна технологія в базисі Галуа – новий напрямок у розвитку комп'ютерних машин. – Львів: CCU'2000. - 2000. – С.31-35.
197. Николайчук Я.М. Архітектура та системні характеристики розподілених комп'ютерних мереж, оснащених асинхронними автономними сенсорами /

Я.М. Николайчук, О.М. Заставний, Н.Д. Круцкевич // Наукові вісті інституту менеджменту та економіки «Галицька академія». – Івано-Франківськ. – 2006. – №2(10). – С. 65-74.
198. Николайчук Я.М. Коди поля Галуа та їх застосування в перетворювачах форми інформації / Я.М. Николайчук, Я.Б. Кусик // 7-й симпозиум: Проблемы создания преобразователей формы информации: тезисы докладов. - Киев: ИКАН Украины, 1992.
199. Николайчук Я.М. Низові обчислювальні мережі: Учбовий посібник.– К:УМК ВО, 1990.– 64с.
200. Николайчук Я.М. Проблеми розвитку методів стиснення масивів даних на основі рандомізації та теоретико-числового базису Галуа / Я.М. Николайчук, І.А. Пилипенко, Н.Я. Возна // Оптико-електронні інформаційно-енергетичні технології. - 2006. - №2(12). - С. 40-47.
201. Николайчук Я.М. Теоретичні основи та інформаційні технології побудови логіко-статистичної інформаційної моделі (ЛСІМ-4) на основі контролю спектральних характеристик об'єктів управління / Я.М. Николайчук, І.В. Андрушко, І.Р. Пітух // Оптико-електронні інформаційно-енергетичні технології.- 2006.- №2(12).- С. 110-118.
202. Николайчук Я.М., Андрушко І.В., Пітух І.Р. Теоретичні основи та інформаційні технології побудови логіко-статистичної інформаційної моделі (ЛСІМ-4) на основі контролю спектральних характеристик об'єктів управління. Міжнародний науково-технічний журнал "Оптико-електронні інформаційно-енергетичні технології".- 2006.- №2(12).- С.110-118.
203. Николайчук Я.М., Возна Н.Я., Пітух І.Р. Проектування спеціалізованих комп'ютерних систем / Навчальний посібник / - Тернопіль: ТзОВ "Терно-граф". 2010. – 392с., іл.
204. Николайчук Я.М., Волинський О.І., Кулина С.В.Теоретичні основи побудови та структура спецпроцесорів в базисі Крестенсона. Вісник Хмельницького національного університету.- Хмельницький.- 2007.- №3.- Т1.- С.85-90.
205. Николайчук Я.М., Воронич А.Р. Багатоканальний спосіб передавання та приймання інформації. Патент на корисну модель № 63648.– 10.10.2011р.
206. Николайчук Я.М., Воронич А.Р. Погонець І.О. Пристрій для визначення автокореляційної міри ентропії. Патент на корисну модель № 58743.– 26.04.2011р.
207. Николайчук Я.М., Гринчишин Т.М., Воронич А.Р. Спосіб передавання і приймання інформації. Патент на винахід № 96853.– 12.12.2011р.
208. Николайчук Я.М., Зевелев С.Я., Крикун З.Н. Выбор оптимальных параметров кодирования методом вычетов. Респ. сб. «Автоматизация и телемеханизация нефтяной промышленности». – 1975. – №2. – С.22-24.
209. Николайчук Я.М., Ищеряков С.М. Метод передачи сообщений в условиях интенсивных помех / Высокопроизводительные преобразователи формы информации и средства передачи данных. Сб.тр. ИК АН УССР. - Киев, 1984, С.62-67.

210. Николайчук Я.М., Круцкевич Н.Д. Перспективи використання зірково - магістральної архітектури з пам'яттю колективного доступу в комп'ютерних мережах з глибоким розпаралелюванням // Вимірювальна та обчислювальна техніка в технологічних процесах: Збірник наукових праць- Хмельницький: ТУП, - 2002. -Т2. - №9. - С. 122 –126.
211. Николайчук Я.М., Круцкевич О.Д. Матричні системи числення. Вісник Хмельницького національного університету.- Хмельницький.- 2007.- №3.- Т1.- С.62-64.
212. Николайчук Я.М., Кусик Я.Б. Коди поля Галуа та їх застосування в перетворювачах форм інформації. Тезисы докладов 7-го симпозиума: Проблемы создания преобразователей формы информации – Киев: ИК АН Украины. – 1992. – С.56-57.
213. Николайчук Я.М., Лучук М.А., Жуган Л.И., Шевчук Б.М Идентификация информационных состояний объектов исследования на основе системы логико–статистических информационных моделей: Препринт/ АН УССР. Ин–т кибернетики им.В.М.Глушкова; 88–45. К.: 1988.- 86 с.
214. Николайчук Я.М., Мельничук С.А. Методи завадостійкого обміну даними в низових обчислювальних мережах автоматизованих систем // 3-я українська конференція „Автоматика 96”.-Севастополь.-1996 –С.34-38.
215. 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.
216. Николайчук Я.М., Стус С.М. Методи вертикальної інформаційної технології в базисі Галуа. 3-я українська конференція “Автоматика 96” Севастополь. - 1996. – С. 77.
217. 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. ИК АН УССР, 1988.
218. Николайчук Я.М., Ширмовский Г.Я., Процюк В.Р. Компактное кодирование сообщений в многоуровневой системе баз данных// Управляющие системы и машины. – 1984. – №1.
219. 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.
220. 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.
221. Николайчук Я.Н. А.С. №754414. – Бюллетень №29. – 1980. Числоимпульсное множильное устройство.
222. Николайчук Я.Н. Автореферат кандидатской диссертации. -Киев: ИК АН УССР, 1979. - 250 с.
223. Николайчук Я.Н. Анализ преобразования базисных функций Радемахера в

АЦП полярного типа, используемого в комплексе СКУБ / Я.Н. Николайчук, Л.А. Гнатив // Автоматизация и телемеханизация нефтяной промышленности. – 1974. - №1. – С. 28-33.
224. Николайчук Я.Н. Аналого-цифровой преобразователь: А.С. № 1372621 Опубл. 07.02.88, Бюл. № 5..
225. Николайчук Я.Н. Аналого-цифровой преобразователь: А.С. № 1462477 Опуб. 28.02.89, Бюл. №8.
226. Николайчук Я.Н. Быстродействующие унитарные вычислите¬ли в Булевом базисе. - Материалы IУ Всесоюзной школы-семинара "Распараллеливание обработки информации". - Львов, ФМИ АН УССР, 1983, С.150-151.
227. Николайчук Я.Н. Взаимосвязь технико-экономических показателей и принципов построения элементов НВС / Я.Н. Николайчук, Т.М. Орищин, М.М. Николайчук – К., 1989. – 80 с. – Деп. в УкрНИИНТИ, №1328.
228. Николайчук Я.Н. и др. Методы реализации протоколов малых локальных вычислительных сетей на основе кодов поля Галуа Локальные вычислительные сети. - Рига: ИЭВТ АН ЛатвССР, 1988, С.
229. Николайчук Я.Н. Кодирование файлов многоуровневой диалоговой системы низовых сетей АСУ / Я.Н. Николайчук // Интерактивные системы: V Всесоюзная школа-семинар: материалы. – Кутаиси, 1983. – С. 87-90.
230. Николайчук Я.Н. Методология формирования, передачи и обработки дискретных сообщений в НВС // Материалы VI Всесоюзной школы-семинара. – Л: ФМИ. – 1987.– С. 187–193.
231. Николайчук Я.Н. Принципы построения и параметры АЦП на основе кодов поля Галуа / Я.Н. Николайчук, Л.Б. Петришин // Проблемы создания преобразователей формы информации: IV Всесоюзн. симпозиум. – Киев. - ИК АН УССР. – 1988. - С. 16-17.
232. Николайчук Я.Н. Реализация узлов специализированных роцессоров на унитарных вычислителях / Методы и средства параллельной обработки информации. Доклады Всесоюзной школы-семинара по распараллеливанию обработки информации. Препринт № 44. Львов, ФМИ, АН УССР, 1981, С. 17-19.
233. Николайчук Я.Н. Цифровые устройства преобразования и обрабитки сигналов с регулярной структурой и высоким уровнем параллелизма операций. - Материалы УI Всесоюзной школы-семинара. - Львов, 1987.
234. Николайчук Я.Н. Шевчук Б.М. Распараллеливание процедуры вычисления мультипликативных функций корреляционной связи. - Материалы УI Всесоюзной школы-семинара. - Львов, 1987, С. 55—56.
235. Николайчук Я.Н. Эффективное кодирование суточного рапорта бурового мастера для формализованного ввода в ЭВМ / Я.Н. Николайчук, Г.Я. Ширмовский // Экспресс-информация: Серия "Экономика и управление в нефтяной промышленности". – 1986. - Вып. 7. – С. 29-33.
236. Николайчук Я.Н., Божнев В.П. Метод уплотнения информации, вводимой в ЭВМ'// Управляющие системы и машины, № 1,1977, С. 108-110.
237. Николайчук Я.Н., 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. - М.: МИНХ и ГП, 1975, С.
238. Николайчук Я.Н., Волошинов С.Д. Распараллеливание информационных потоков и повышение живучести НВС на основе многопортовой памяти коллективного пользования/ Распараллеливание обработки информации, кн.3.- Львов, ФМИ, 1989, с.86-88.
239. Николайчук Я.Н., Гнатив Л.А. Анализ преобразования базисных функций Радемахера в АЦП полярного типа, используемого в комплексе СКУБ // Автоматизация и телемеханизация в нефтяной промышленности. - М.:. ВНИИОЭНГ, вып. 12, 1984, с.4-7.
240. Николайчук Я.Н., Доценко Р.В., Петришин Л.Б. Принцип параллельной передачи сообщений в НВС на базе весового суммирования сигналов и кодов поля Галуа. - Материалы УІ Всесоюзной школы-семинара. - Львов, 1987.
241. Николайчук Я.Н., Ищеряков С.М. Параллельная свертка отсчетов в унитарных вычислителях статистических оценок. - .Материалы ІУ Всесоюзной школы-семинара "Распараллеливание обработки информации". - Львов ФМИ АН УССР, 1983, С.152-153.
242. Николайчук Я.Н., Ищеряков С.М А.С. №1115062.-Бюллетень №35.-1984. Многоканальное устройство для вычисления модульной функции.
243. Николайчук Я.Н., Ищеряков С.М. А.С. №1317455.-Бюллетень №22.-1987. Многоканальное устроиство для вычисления функций эквивалентности.
244. Николайчук Я.Н., Ищеряков С.М. Перспективы использо-вания многоканальных унитарных вычислителей статистических функций в читающих автоматах. Тезисы докладов Y Всесоюзной конференции: Автоматизация ввода письменных знаков в ЦВМ.-Каунас.-1984
245. Николайчук Я.Н., Ищеряков С.М. Цифровые устройства свертки и корреляционной обработки широкополосных сигналов в реальном времени. - Тезисы докладов Всесоюзной конференции Методы и микроэлектронные средства цифрового преобразования и бработки сигналов. - Рига, 1986, с.
246. Николайчук Я.Н., Ищеряков С.М., Шевчук Б.М., Кусык Я.Б. Вычислительная среда для цифровой фильтрации широкополосных сигналов / Методы и микроэлектронные средства цифровой обработки и преобразования сигналов". Рига: ИЭТВ, 1989, с.260-261.
247. Николайчук Я.Н., Ищеряков С.М.А.С. №1280394.-Бюллетень №48. – 1986. Многоканальное устройство для вычисления модульной функции.
248. Николайчук Я.Н., Клим Б.В. Формализация процедуры ввода в ЭВМ суточного рапорта бурового мастера. // Автоматизация и телемеханизация нефтяной промышленности, № 6, 1981, С. 20-23.
249. Николайчук Я.Н., Лучук М.А., Шевчук Б.М., Жуган Л.И. Идентификация информационных состояний объектов исследования и управления на основе системы логико-статистических информационных моделей/ Препр. АН УССР. Ин-т кибернетики им. В.М.Глушкова; Киев, 1988. - 20 с.
250. Николайчук Я.Н., Петришин Л.Б. Принципы построения и параметры АЦП на основе кодов поля Галуа // Проблемы создания преобразователей формы информации. Тезисы докладов IV Всесоюзного симпозиума.-Киев, ИК АН

УССР, 1988, С.16-17.
251. Николайчук Я.Н., Петришин Л.Б., Романюк Ю.Ф. и др. Нормирование и дистанционный контроль скорости вращения счетчиков электрической энергии с использованием кода Галуа. Рук. деп. в Укр.НИИНГИ № 1092, 1989. - 19 с.
252. Николайчук Я.Н., Петришин Л.Б., Турчанинов Ю.Н., Волошинов С.Д. Звездно-кольцевая вычислительная система с коллективной памятью многопроцессорного доступа / Материалы III Всесоюзного симпозиума: Перспективы развития вычислительных систем. -.Рига, РПИ, 1989, с.22.
253. Николайчук Я.Н., Петришин Л.Б., Шевчук Б.М. Сжатие данных на основе кодов поля Галуа / Пробелмы создания преобразователей формы информации. Тезисы докладов УI Всесоюзного симпозиума. - Киев, ИК АН УССР, 1988, с.18-19.
254. Николайчук Я.Н., Пицык В.Л. Алгоритм ускоренного вычисления автокорреляционной функции. - Рук. деп. в УКРНИИНТИ, №1556, 1988. - 5 с.
255. Николайчук Я.Н., Ролик В.А., Зевелев С.Я., Савин З.В., Доценко Р.В. Метод весового уплотнения и селекции измерительных каналов с передачей по ВОЛС. - Тезисы докладов Всесоюзной научно-технической конференции по функциональной оптоэлектронике "Оптоэлектронные методы и средства обработки информации". - Винница, 1987 .
256. Николайчук Я.Н., Ролик З.А., Ишеряков С.М., Петришин Л.Б. Новые принципы и технические средства контроля забойных параметров бурения на основе кодов поля Галуа. - Тезисы окладов Областной научно-технической конференции "Состояние перспективы геолого-геофизических исследований, проводимых процессе бурения скважин." - Тюмень, 1987.
257. Николайчук Я.Н., Шевчук Б.М. Методы цифровой обработки шумоподобных сигналов на основе кодовых ключей. - В кн. Технические средства обработки информации для высокопроизводительных ЭВМ и систем. - Киев, Сб.,научн.тр. ИК АН УССР, I988,с.26-28.
258. Николайчук Я.Н., Шевчук Б.М. Реализация физического уровня НВС на основе цифровых приемопередатчиков с квазитроичной манипуляцией. - Тезисы.докладов Всесоюзной научно-технической конференции "вычислительные сети коммутации пакетов". - Рига: Институт электроники и вычислительной техники АН ЛССР, 1987.
259. Николайчук Я.Н., Шевчук Б.М., Попов А.А. Функции взаимокорреляционной связи и их применение для анализа биомедицинских исследований. - Тезисы докладов 2-й Всесоюзной конференции "Технические средства для заболеваний ССС", - Киев, 1987.
260. Николайчук Я.Н., Ширмовский Г.Я. Рациональное кодирование и концентрация данных в низовой сети АСУ ТП бурение. // Автоматизация и телемеханизация нефтяной промышленности. - М., ВНИИОЭНГ, № 3, 1983с.
261. Николайчук Я.Н., Ширмовский Г.Я. Быстродействующая рекурсивная процедура извлечения вычетов. - Материалы IУ Всесоюзной школы-семинара "Распараллеливание обработки информации" - Львов,. ФМИ АН УССР, 1983, С. 92-93.

262. Николайчук Я.Н., Ширмовский Г.Я. Методы распараллеливания операций в низовых вычислительных сетях на базе преобразования СОК. - Распараллеливание обработки информации. - Доклады Всесоюзной школы-семинара. Препринт АН УССР, 1985.
263. Николайчук Я.Н., Ширмовский Г.Я. Многоканальная система волоконно-оптической связи с уплотнением в унитарном коде СОК. - Тезисы докладов Всесоюзного семинара "Оптоэлектронные устройства в приборостроении и информатике". - Тбилиси,1985, С.
264. Николайчук Я.Н., Ширмовский Г.Я. Эффективное кодирование суточного рапорта бурового мастера для формализованного вода в ЭЗМ. - Экспресс-информация. Серия "Экономика и управление в нефтяной промышленности", вып. 7, 1986, с.29-33.
265. Николайчук Я.Н., Ширмовский Г.Я., Кукурудз С.Ф. Программные модели распараллеливания измерения кодирования и передачи сообщений унитарным преобразованием СОК. - Материалы УІ Всесоюзной школы-семинара. - Львов, 1987.
266. Николайчук Я.Н., Ширмовский Г.Я., Петришин Л.Б. Кодирование сообщений, для создания локальной базы данных по разведочным скважинам / НТИБ Серия "Нефтегазовая гешогия, геофизика и бурение. - М., ВНИИОЭНГ, 1984, вып. 7, С.
267. Николайчук Я.Н., Ширмовский Г.Я., Процюк В.Р. Компактное кодировние сообщений в многоуровневой системе баз данных // Управляющие системы и машины, № 1 , 1984, С.102-105.
268. Николайчук Я.Н.,Методы распараллеливания операций и накопителях большой емкости. - Распараллеливание обработки информации. - Доклады Всесоюзной школы-семинара. - Препринт №44. - Львов, ФМИ АН УССР, 1985.
269. 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.
270. Николайчук Я. Н. Функции взаимокорреляционной связи и их применение для вычисления структурной функции. Тезисы докладов II Всесоюз.конф. «Технические средства для диагностики заболеваний ССС».-Москва.-1987.
271. Николюк О.М., Жуган Л.И., Николайчук Я.Н. Логическое сканирование системы импульсных источников Галуа / Тезисы докладов XI Всесоюзного симпозиума: Логическое управление с использованием ЭВМ. - М.: МГИ, 1990, с.339-341.
272. Никонов В.Ф. и др. Систолическая обработка информации элементная база и алгоритмы // Зарубежная радиоэлектроника, 1987, № 7, с.34-51.
273. Никульский И. Оптические интерфейсы цифровых коммутационных станций и сети доступа.М.: Техносфера, 2006.-563 с.
274. Новиков Ю., Новиков Д., Черепанов А., Чуркин В. Компьютеры, сети, Интернет. Энциклопедия.– СПб.: Питер, 2002.– 928с.

275. Овчаров Л.А., Селетков С.Н. Автоматизированные банки данных. - М.: Финансы и статистика, 1982. - 262 с.
276. Озкарахан З. Машины баз данных и управление базами даных: Пер. с анг.- М.: Мир, 1989.- 696 с.
277. Окунев Ю.Б., Яковлев А.А. Широкополосные системы связи с составными сигналами. Под ред. А.М. Заездного. - М.: Сов. радио, 1968 - 168 с.
278. Ольховский Ю.Б., Новоселов О.Н., Мановцев А.П. Сжатие данных при телеизмерениях. - М.: Советское радио, 1971.- 340 с.
279. Оокоси Т. Оптоэлектроника и оптическая связь. Пер. с япон. - М.: Мир, 1988. - 96 с.
280. Осипов Д. Delphi Профессиональное программирование. – М.: Россия, 2004. – 1056 с.
281. Остапенко А. Г., Лавлинский С. И., Сушков А. В. и др. Цифровые процессоры обработки сигналов: Справочник. / Под ред. А. Г. Остапенко. - М.: Радио и связь, 1994. - 264 с.
282. Палагин А.В., Жаровский С.Н., Павлишин С.В. Организация и принципы построения локальной сети управления техноло¬гическими объектами. - Киев: ИК АН УССР, 1987. - 23 с.
283. Петришин Л. Б. Цифровая обработка сигналов на основе преобразования кодов поля Галуа / Л. Б. Петришин, Я. Н. Николайчук, С. М. Ищеряков // Методы и микроэлектронные средства цифрового преобразования и обработки сигналов. – Рига: ИЭВТ АН ЛатвССР, 1989. – Т. 1. – С. 130–132.
284. Петрович И.Т., Размахнин М.К. Системы связи с шумоподобными сигналами. – М.: Советское радио, 1969. – 232 с.
285. Петрович Н.Т., Размахнин М.К. Системы связи с шумоподобными сигналами. - М.: Сов.радио, 1969. - 180 с.
286. Петропавловский В.П., Синицин Н.В. Фазовые цифровые преобразователи угла. - М.: Машиностроение, 1984. - 136 с.
287. Писаренко В. Ф. Выборочные свойства спектральной оценки максимальной энтропии // Вычислительная сейсмология, 1977. Вып. 10. – С. 118–149.
288. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. – М.: Мир, 1976. - 400с.
289. Пітух І. Особливості структурної організації фреймів в комп'ютерних мережах з глибоким розпаралеленням інформаційних потоків// Вісник Технологічного університету Поділля . –Хмельницький.- 2005.- №4,Ч.1, Т.2. С.7–10.
290. Пітух І. Інформаційна технологія побудови миттєвих та інтегральних економічних епюр руху даних на основі циклів матричних моделей комп'ютерних систем //Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2007. – Т.1, №3. – С.130-134.
291. Пітух І. Кореляційні та ентропійні моделі об'єктів управління розподілених комп'ютерних мереж// Наукові вісті інституту менеджменту та економіки "Галицька академія". Ів. Франківськ.–2006. – № 2 (10).– С.117 – 120.
292. Пітух І. Критерії ефективності використання ресурсів архітектури інформаційних систем, які реалізують моделі руху даних // Вісник

Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2006. – №5.– С.106-109.
293. Пітух І. Проектування характеристик системних об'єктів комп'ютерних мереж з глибоким розпаралеленням інформаційних потоків // Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2005. – Т.2, Ч.1, №4. – С.133-136.
294. Пітух І., Николайчук Я., Возна Н. Моделювання руху даних та методологія проектування комп'ютерної мережі з паралельними інформаційними потоками // Вісник Технологічного університету Поділля. Технічні науки. – Хмельницький. – 2004. – Т.2, Ч.1, №2.– С. 33-35.
295. Пітух І.Р. Моделі комп'ютерних мереж на основі інтегральних економічних епюр // Збірник наукових праць, Інститут проблем моделювання в енергетиці НАН України. – Київ. – 2004.– № 27. – С.81–86 .
296. Пітух І.Р. Теоретичні основи побудови моделей економічних епюр руху даних в комп'ютерних мережах з використанням різних теоретико – числових базисів // Збірник наукових праць. Інститут проблем моделювання в енергетиці НАН України. – 2006.– № 37. – С.42–46 .
297. Полонников Р.И. Матричные методы обработки сигналов / Полонников Р.И., Костюк В.И., Краскевич В.Е. - К.: Техніка, 1977. – 136с.
298. Прангишвили И.В. Микропроцессоры и локальные сети микро-ЭВМ в распределенных системах управления. - М. : Энергоиз-дат, 1985. – 272 с.
299. Прэтт У. Цифровая обработка изображений: в 2-х кн. / Прэтт У.; пер. с англ. Д. С. Лебедева. – М.: Мир, 1980.
300. Погонець І.О., Воронич А.Р. Ентропійні характеристики одновимірних та двовимірних ШПС. Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2011. - №7. Т1. – С.128-131
301. Пуртов С.Т. Автоматизированные системы управления предприятием.–М.: Высшая школа, 1989.–396с.
302. Пухов Г.Е., Евдокимов В.Ф., Синьков М.В. Разрядно- аналоговые вычислительные системы. - М.: Сов.радио, 1978. - 256 с.
303. Радченко А.Н., Филиппов В.И. Сдвигающие регистры с логической обратной связью и их использование в качестве счетных и кодирующих устройств // Автоматика и телемеханика, 1957, № II, с.
304. Ракошиц В. С., Козлов В. П., Можаев И. А. Специализированные микропроцессоры, реализующие быстрые преобразования // Цифров. обраб. сигналов и её применение. М., 1981. – С. 38-56.
305. Раскин Д. Интерфейс: Новые направления в проектировании компьютерных систем - СПб.: Символ 2005.-272с.
306. Распределенные системы управления за рубежом // ТС-3 Автоматизированные системы управления, вып.1. - М.: ИНФОРМ- ПРИБОР, 1988. - 10 с.
307. Ратхор Т.С. Цифровые измерения. АЦП/ЦАП: Перевод с английского. М., Техносфера, 2006 – 856с.
308. Романенко А.Ф., Сергеев Г.А. Вопросы прикладного ана лиза случайных процессов. - М.: Сов.радио, 1965. - 255 с.

309. Романец Ю.В., Тимофеев П.А., Шангин В.Ф. Защита информации в компьютерных системах и сетях.-М.:Радио и связь, 1999.-328с.
310. Романов В.А. Элементная база – основа интеллектуализации средств вычислительной техники / В.А. Романов // Компьютерные средства, сети и системы: Сб. научных трудов. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2002. – №1. – С. 20-26.
311. Садыхов Р.Х., Чеголин П.М., Шмерко В.П. Методы и средства обработки сигналов в дискретных базисах. – Мн.: Наука и техника, 1987. - 296 с.
312. Саломаа А. Криптография с открытым ключом/Пер.с англ.-М.:Мир, 1996.-318с.
313. Сегін А. І. Оцінка впливу старіння інформації на кореляційні моделі багатоканальних об'єктів управління // Розвідка і розробка нафтових і газових родовищ. Серія: Технічна кібернетика та електрифікація об'єктів паливно-енергетичного комплексу. № 36. Т.6 – Івано-Франківськ.: ІФДТУНГ, 1999. – С. 80-88.
314. Сегін А. Ідентифікація двовимірних зображень на основі інформаційної моделі хемінгового простору // Комп'ютерні технології друкарства. Збірник наукових праць. – Львів, – № 4, – 2000. – С. 344–347.
315. Сеньо П.С. Теорія ймовірностей та математична статистика : Підручник.–К : Центр навчальної літератури , 2004. – 448 с.
316. Сергиенко А. М. VHDL для проектирования вычислительных устройств, К., ТИД "ДС", 2003,- 2008с.
317. Серебренников М.Г., Первозванский А.А. Выявление скрытых периодичностей. - М.: Наука, 1965. - 244 с.
318. Силкин Л.Б. Малогабаритный цифратор перемещений // Электросвязь, 1976, № 6, с.48-49.
319. Синьков М.В. Развитие методов теории чисел для расширения возможностей вычислительных средств, функционирующих системе остаточных классов. - В кн.: Проблемы электроники и моделирования. - Киев: Наукова думка, 1976.
320. Сипсер Р. Архитектура связи в распределительных системах. Т. I, 2. - М.: Мир, 1981. - 744 с.
321. Системы обработки информации. Защита криптографическая, Алгоритм криптографического преобразования ГОСТ 28147-89.
322. Системы параллельной обработки: Пер. с англ. / Под ред. Д.Ивенса. - М.: Мир, 1985. - 416 с.
323. Сингер.М, Берг. П. Гены и геномы. – М.: Мир. – 1998. – 360 с.
324. Системы передачи данных и сети ЭВМ // ТИИЭР, 1972, т.60, № II. - 220 с.
325. Системы передачи данных и сети ЭВМ. - М.: Мир, 1974. - 215 с.
326. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е изд.: Пер. с англ. – М.: Издательський дом "Вильямс", 2003. - 1104с.
327. Смолов В. Б. Функциональные преобразователи информации – Л.: Энергоиздат, 1981. – 248 с.

328. Смолов В. Б., Фомичев В.С. Аналого-цифровые и цифро-аналоговые нелинейные вычислительные устройства. – Л.: Энергия, 1974. – 336 с.
329. Советов Б.Я. Моделирование систем. Практикум: Учебное пособие .- М.- Высш. шк., 2005.- 295 с.
330. Советов Б.Я.и др. Применение микропроцессорных средств в системах передачи информации. - М.: Высш.школа, 1987.-256 с.
331. Спилкер Дж. Цифровая спутниковая связь; Пер. с англ / Под ред. В.В.Маркова. - М.: Связь, 1969.- 92 с.
332. Стешенко В. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов – М.: «Додека», 2000. – 224с.
333. Стил Р. Принципы дельта-модуляции. – М: Мир, – 1979. – 219 с.
334. Стус С.М., Николайчук Я.Н. Вертикальное программирование параллельных процессов в распределенных вычислительных системах/ Распараллеливание обработки информации, кн.1. - Львов: ФМИ, 1989, с.222-223.
335. Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL, Санкт-Петербург, BHV,- 2003, 576с.
336. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон; пер. с англ. В. В. Чепыжова. – М.: Техносфера, 2004. – 368 с.
337. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон; пер. с англ. В. В. Чепыжова. – М.: Техносфера, 2004. – 368 с.
338. Таллер В. Теоретические ограничения скорости передачи информации. Теория информации и её приложения / Под ред. А. А. Харкевича – М.,: Физматгиз, 1959. – С. 58–81.
339. Тарасов И.Е. Разработка цифровых устройств на основе ПЛИС XILINX с применением языка VHDL. М.: Горячая линия-Телеком, 2005. – 782с.
340. Толстяков В.С. Обнаружение и исправление ошибок в дискретных устройствах. - М.: Сов.радио, 1972. - 287 с.
341. Томашевський В.М. Моделювання систем.– К.: Видавнича група BHV, 2005.– 352с.
342. Торгашев В.А. Система остаточных классов и надежность ЦВМ / Торгашев В.А. - М.: Советское радио, 1970. – 118 с.
343. Тутевич В.Н. Телемеханика. - М.: Энергия, 1973. - 297 с.
344. Угрюмов Е. П. Цифровая схемотехника. – СПб.: БХВ – Санкт-Петербург, 2002. – 528 с.
345. Управляющие вычислительные машины в АСУ технологическими процессами /Под. ред. Т. Харрисона.- Т.1. – М.: Мир, 1975.– 230с.
346. Управляющие вычислительные машины в АСУ технологическими процессами /Под. ред. Т. Харрисона/- Т.2. – М.: Мир, 1975.– 530с.
347. Урядников Ю. Ф., Аджемов С. С. Сверхширокополосная связь. Теория и применение. – М.: «СОЛОН-Пресс», 2005. – 368 с.
348. Фараджев Р.Г., Цыпкин Я.З. Преобразование Лапласа-Галуа в теории последовательностных машин // Докл. АН СССР, 1966, № 3, с.45-52.
349. Фаронов В.В. Delphi: Программирование на языке высокого уровня. – К.: Вища школа, 2006 – 640 с.

350. Федорков Б. Г., Телец В. А.Микросхемы ЦАП и АЦП: функционирование, параметры, применение. — М,: Энергоатомиздат, 1990. — 320 с: ил.
351. Федорков Б.Г. и др. Микроэлектронные цифро-аналоговые и аналого-цифровые преобразователи. - М.: Радио и связь, 1984. - 120 с.
352. Феррари Д. Оценка производительности вычислительных систем. – М.: Мир, 1981. – 576 с.
353. Франк-Каменецкий М.Д. Самая главная молекула. М.: Наука. 1983. – 160 с.
354. Хаусли Т. Передача данных и системы телеобработки. - М.: Радио и связь, 1982. - 200 с.
355. Хаусли Т. Системы передачи и телеобработки данных: Пер. с англ.– М.: Радио и связь, 1994.–456 с.
356. Хетагуров Я.А. Проектирование автоматизированных систем обработки информации и управления.: М.-Висш.шк.-2006.-223 с.
357. Хетагуров Я.А., Руднев Ю.П. Повышение надежности цифровых устройств методами избыточного кодирования. - М.: Энергия, 1975. - 280 с.
358. Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы. М.: Радио и связь. 1986. 392 С.
359. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем : Учебник для вузов .–СПб.: Питер, 2006.– 668 с.
360. Чайковский В.И. Функциональные особенности цифровых анализаторов спектра, работающих в реальном масштабе времени. – Киев.: (Препр. / Ин-т кибернетики АН УССР;76-39), 1976. – 32 с.
361. Чеголин П.М. Автоматизация спектрального и корреляционного анализа. - М.: Энергия, 1969. - 383 с.
362. Червяков Н.И. Нейрокомпьютеры в остаточных классах: учеб. пос. для вузов / Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н.; под ред. А.И. Галушкина. - М.: Радиотехника, 2003. – Кн. 11. - 272 с. – (Научная серия «Нейрокомпьютеры и их применение).
363. Чердынцев В.А. Проектирование радиотехнических систем со сложными сигналами. - Минск: Вышэйшая школа, 1979. - 192 с.
364. Черкаський М.В. Мурад Хусейн Халіл. Універсальна SH-модель // Комп'ютерні системи та мережі: Вісник Національного університету "Львівська політехніка". – Львів, 2004.-№523.-С.150-154.
365. Черкаський М.В., Мурад Х.Х. Складність пристрою керування// Комп'ютерна інженерія та інформаційні технології Вісник Національного університету "Львівська політехніка". – Львів, 2004.-№521.-С. 3-7.
366. Черкаський М.В., Мурад Хусейн Халіл. Аналіз складності пристроїв помноження//Комп'ютерні системи проектування. Теорія і практика: Вісник Національного університету "Львівська політехніка" – Львів, 2005. - №548. – С.15-21.
367. Чирка М.І. Метод побудови розподіленого температурного сенсора на основі системи числення базису Крестенсона.// Чирка М.І., Касянчук М.М., Якименко І.З./ Проблемно-наукова міжгалузева конференція «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделю-вання та управління ПНМК-2011, Україна, Бучач 17-20 травня 2011

року.
368. Число и мысль / Сборник работ, вып. 9. - М.: Знание, 1986. - 176 с.
369. Шандровська Л.М., Возна Н.Я. Реалізація цифрового підпису в телекомунікаційних системах та його правові аспекти//Вісник технологічного ун-ту «Поділля».-2003.-Т1.-№3.-С.125-127.
370. Широчин В.П., Ху Чженбин. Средства сопровождения адаптивных комплексных систем защиты информации // Сб. трудов IV междунар. научно-практической конф. "Проблемы внедрения информационных технологий в экономике и бизнесе". – Ирпень. - 2003. – С. 661 – 664.
371. Шишка Р.Б. Охорона права інтелектуальної власності: авторсько-правовий аспект.- Харків: Ви-во Нац. Ун-ту внутр. справ, 2002.- 368с.
372. Шляндин В.М. Цифровые измерительные устройства. - М.: Высшая школа, 1981. - 288 с.
373. Шнейдер Ю.А., Шаров А.А. Системы и модели.– М.:Радио и связь, 1982.– 348 с.
374. Шпак Ю.А. Разработка приложений в Delphi 2005/2006.- М.: МК-Пресс, 2006.- 544 с.
375. Штарьков Ю.М. Адаптивное кодирование дискретных источников. – В кн.: Кодирование в сложных системах. – М.: Наука, 1974. С. 164-169.
376. Шушков Е.И., Цодиков М.Б. Многоканальные аналого- цифровые преобразователи. - Л.: Энергия,1975. - 160 с.
377. Ямпольский Л.С., Брунштейн Ю.Г. Преобразователи линейных перемещений. - Киев: Техніка, 1984. - 144 с.
378. Ярославский Л.П. Введение в цифровую обработку изображений. - М.: Сов.радио, 1979. - 312 с.
379. 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.
380. 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.

Іноземна література

1. ByteBlaster MV Parallel Port Download Cable, Data Sheet, Altera corporation, ver.1, April, 1998.
2. Cunningham D., Lane W.G. Gigabit Ethernet Networking.- Macmillan Technical Publishing, 1999.
3. Curran T. Folkess 6800 peripheral chips assume 1/0 tasks and mor //Electron. Des.- 1983.-vol. 51, № 21 – p. 123-128.
4. Digital Signal Processing Applications Using the ADSP-2100 Family, Vol. 1 and Vol. 2, Analog Devices, Free Download at: http://www.analog.com.
5. DSP Designer's Reference (DSP Solutions) CDROM // Analog Devices, 1999.

6. DSP Navigators: Interactive Tutorials about Analog Devices' DSP Architectures (Available for ADSP-218x family and SHARC family) : http://www.analog.com/industry/dsp/training/index.html.
7. Edwards T.C. Foundations of Interconnect and Microstrip Design, .- John Wiley and Sons, 2000.
8. Henry Ott. Noise Reduction Techniques in Electronic System.- John Wiley & Sons, 1988.
9. Pitukh, Y. Nykolaychuk, N. Vozna. Information technologies of models data movement construction in the automatic management systems // Proc. of the VIIII–th International Conf. CADSM 2005. – Lviv–Polyana (Ukraine). – 2005.- P.427–428.
10. Kaiser J. F. Design Methods for sampled Data Filters, Proc. First allerton Conf. An Curcuit and System Theory, 221-236(Nov.1963).
11. Keh-La Lin, Armin Kemma and Berdich J. Hosticka Modular low-power high-speed CMOS analog-to-digital converter for embedded systems. Kluwer Academic Publishers, Boston, 2003, - 254c.
12. Kung H Systolic algoritms for the CMN WARP Processor//Int. Conf. Pattern. Recgn. Monreal, July, 30-august 2, 1984, p.570-577.
13. Longo G. Quantitative-qualitattive measure of information. Internat. Centre of mechan. Sciences (Sommerkurs in Undien). Springer-Verlag, 1972.
14. Lyubov Nykolaychuk, Oksana Chehodar.- Problems in Creation of Systems of Knowledge and Estimation of Entropy of Legal Information. – Modern Problems of Radio Engineering, Telecommunications and Computer Science. – Proceedings of the International Conference TCSET'2006. – Lviv-Slavsko, Ukraine February 28 – March 4, 2006.
15. M. Dyvak, Yu.Franko, I.Pituh, S. Voloshchyk. The full combination algorithm modification in the task of technological process interval modeling// Proc. the VI–th International Conf. CADSM 2001.- Lviv–Slavsko (Ukraine).-2001. – P. 133.
16. M. Dyvak, Yu.Franko, I.Pituh, V. Tsymbaliy. Algorithm of technological process interval modeling// Proc. International Conf. on Modern Problems of Telecommunications, Computer Science and Engineers Training.- Lviv–Slavsko, (Ukraine)-2000. – P. 31.
17. M.K. Simon, 'Spread Spectrum Communications Handbook', Me Graw-Hill, Inc., 1994 – 238c.
18. Neto J.P., Siegelmann H.T., COSTA J.F., ARAUJO C.P.S. Turing Universality of Neural Nets (revisited). // Lecture Notes in Computer Science- 1333, Springer-Verlag. - 1997. - P. 361-366.
19. Nykolaychuk Y. The theory of designing specialized computer systems on the basis of analogy objects of power system / Y.Nykolaychuk, A.Segin, N.Krutckevych, N.Vozna // Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці: VII міжнар. наук.-техн. конф. CADSM'2003: матеріали. – Львів-Славсько, 2003.- С 241-243.
20. Nykolaychuk Y., Yatskiv N., The coding of multichannel sources of information //Proc. of the VIIth International Conf. The Experience of Designing and Application

of CAD Systems in Microelectronics. CADSM 2003. – Lviv-Slavske (Ukraine).– 2003. – P. 249 – 250.
21. Patrick J Zabinski. Surface roughness of pcb tracks…// Correspondence to the SI–LIST.- 11 Jun 2001.
22. Patrick J Zabinski. Surface roughness of pcb tracks…// Correspondence to the SI–LIST.- 11 Jun 2001.
23. Pening P.J., Buzen J.P. Operetional analysis of queueing network models //Computing Serveys.– 1978.– Vol. 10, № 3.– P.225–261.
24. Pitukh I, Nykolaychuk Y., Vozna N.. Principles of computer networks construction with deep paralleling of information flows on the basis of matrix models of data movement // Proc. of International Conf. "Modern Problems of Radio Engineering, Telecommunicatioons and Computer Science". (TCSET'2004). Lviv-Slavske (Ukraine).-2004.- P.417 – 419.
25. Pitukh, Y. Nykolaychuk, N. Vozna. Information technologies of models data movement construction in the automatic management systems // Proc. of the VIIII–th International Conf. CADSM 2005. – Lviv–Polyana (Ukraine). – 2005.- P.427–428.
26. Ralph D., Graham P. MMS: Technologies, Usage and Business Models.: Пер. с англ.. – М.: Издательский дом «Вильямс», 2003. – 986 с.
27. Ralph D., Graham P. MMS: Technologies, Usage and Business Models.: Пер. с англ.. – М.: Издательский дом «Вильямс», 2003. – 986 с.
28. Reed I.S., Troung T.K. Complex Integer convolutions over direct sum Galois fields. – IEEE Trans. Inter., 1975, 21 p. 647-661.
29. Shore J.E. Second Thoughts on Parallel Processing // Comput. - Elect. Eng. - N1. - P.95-109.
30. Shults G. Informationstheorie mit Bewertung. Wiss/ Zeitschrift d. Humbold Univer. Berlin XX – 1971. S. 175-183.
31. Svensson C., Dermer G. Time Domain Modeling of Lossy Interconnect//, IEEE Trans. Advanced Packaging.-2001.- Vol. 24, №. 2.
32. Svensson C., Dermer G. Time Domain Modeling of Lossy Interconnect//, IEEE Trans. Advanced Packaging.-2001.- Vol. 24, №.2.
33. Voloshinov S, Nicolaychuk J. Parallel acess processec in the starring computer network/ The First international conference an information technologies for image analysis and pattern recognition.-Lviv: FMI, 1990, p.212-218.
34. Y. Nykolaychuk, N. Krutskevych O. Zastavniy, T. Grinchyshyn, Perspective Architecture and Components of Computer Networks // Proc. Of the Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS, Lviv, Ukraine, 2003.
35. Y. Nykolaychuk, N. Krutskevych, O. Zastavnyy Architecture and system characteristic of distributed computer network with autonomus sensor equipment// Proc. of the International Conf."Modern problems of radio engineering, telecommunications and computer science" TCSET 2006. – Lviv-Slavsko (Ukraine). – P. 394 – 398c.
36. Y.Nikolaychuk, I.Andrushko. Theoretical Bases of Logical Statistic Informative Models and Prospect of Their Application in the Distributed Computer System//

Матеріали VIII Міжнар. наук.–техн. конференції CADSM 2005.- Львів-Поляна.- 2005.-265с.
37. Yaroslaw Nykolaychuk. Igor Pitukh, Natalia Vozna, Lyubov Nykolaychuk.- Information technologies of models formalization and designing for data movement in computer networks of automatic control system.- Proceeding of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: IDDAACS‘2005.-2005.-P.253-259.
38. Zastavniy O. Analog-digital Coders in Galois Base // Proc. of the International Conf. CADSM 2005. - Lviv-Slavsko (Ukraine). – 2005, - P. 248.
39. Zhugan L., Ishetyakov s., nikolaythuk J., Petryhyn L. Recognition of comlicated passive obiect models in terms hight-duty parallel processors/ informatoin technologies for image analysis and pattern recognition ITIAPR’90. The First international Conference, - Lviv: FMI, 1990 Volume II, p.224-227.
40. Gatlin L.L., Information Theory and the Living System, Columbia University Press, New York, 1972.
41. Mac Donaill D.A., “A parity code interpretation of nucleotide alpha-bet composition,” Chemical Communications, pp. 2062-2063,2002.
42. Schneider T.D., “Some lessons for molecular biology from information theory,” Entropy Measures, vol. 119, pp. 229-237,2003.
43. May E.E. et. al., “Coding theory based models for protein translation initiation in prokaryotic organisms,” in Fifth International Workshop on Information Processing in Cells and Tissues (IPCAT), September 2003.
44. Rosen G.L. and Moore J.D., “Investigation of coding structure in dna,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2003.
45. Burdon R.H., Genes and the Environment, Taylor and Francis, Inc., Pennsylvania, 1999.
46. Anastassiou, “Genomic signal processing,” IEEE Signal Processing Magazine, July 2001.
47. http://opal.biology.gatech.edu/GeneMark/.
48. Wicker S.B., Error Control Systems, Prentice Hall, New Jersey, 1995.
49. Golub G.H. and Van Loan C.F., Matrix Computations, The Johns Hopkins University Press, Maryland, 1996.
50. Hauth A., Identification of Tandem Repeats Simple and Complex Pattern Structures in DNA, Ph.D. thesis, University of Madison, Wisconsin, 2002.
51. Kolpakov R., Bana G., and Kucherov G., mreps, http://www.loria.fr/mreps/.
52. M. Arita and S. Kobayashi. DNA sequence design using templates. New Generation Computing, vol. 20 (2002), 263-277.
53. G. T. Bogdanova, A. E. Brouwer, S. N. Kapralov, and P. R. J. Ostergard. Error-correcting codes over an alphabet of four elements. Designs, Codes and Cryptography, vol. 23 (2001), 333-342.
54. E. Brouwer. Bounds on the size of linear codes. In Handbook of Coding Theory (editors V. S. Pless and W. C. Huffman), North-Holland, 1998, pp. 295-461.

55. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith. A new table of constant weight codes. IEEE Trans. Inform. Theory, vol. 36 (1990), 1334-1380.
56. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction via codes over GF(4), IEEE Trans, Inform. Theory, vol. 44 (1998) 1369-1387.
57. J. H. Conway and N. J. A. Sloane. Lexicographic codes: error-correcting codes from game theory. IEEE Trans. Inform. Theory, vol. 32 (1986), 337-348.
58. T. Etzion. Optimal constant weight codes over Zk. and generalized designs. Discrete Math., vol. 169 (1997), 55-82.
59. D. Faulhammer, A. R. Cukras, R. J. Lipton, and L. F. Landweber. Molecular computation: RNA solutions to chess problems. PNAS, vol. 97 (2000), 1385-1389.
60. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith and R. M. Corn. Demonstration of a word design strategy for DNA computing on surfaces. Nucleic Acids Research, vol. 25 (1997), 4748-4757.
61. G. Höhn. Self-dual codes over the Kleinian four group. Preprint, available electronically at arXiv:math.C0/0005266 .
62. S. M. Johnson. A new upper bound for error-correcting codes. IRE Trans. Inform. Theory, vol. 8 (1962), 203-207.
63. S. M. Johnson. Upper bounds for constant weight error-correcting codes. Discrete Math., vol. 3 (1972), 109-124.
64. S. Kobayashi, T. Kondo, M. Arita. On template method for DNA sequence design. In DNA Computing: 8th International Workshop on DNA-Based Computers (editors M. Hagiya and A. Ohuchi), Springer LNCS vol. 2568, 2002, pp. 205-214.
65. M. Li, H. J. Lee, A. E. Condon, and R. M. Corn. DNA word design strategy for creating sets of non-interacting oligonucleotides for DNA microarrays. Langmuir, vol. 18 (2002), 805-812.
66. S. Litsyn. An updated tables of the best binary codes known. In Handbook of Coding Theory (editors V. S. Pless and W. C. Huffman), North-Holland*, 1998, pp. 463-498.
67. F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correctinq Codes, North Holland, 1977.
68. Marathe, A. E. Condon, and R. M. Corn. On combinatorial DNA word design. Journal of Computational Biology, vol. 8 (2001), 201-220.
69. K. U. Mir. A restricted genetic alphabet for DNA computing. In DNA Based Computers II (editors L. F. Landweber and E. B. Baum), AMS/DIMACS, 1999, pp. 243-246.
70. P. R. J. Östergärd and M. Svanström. Ternary constant weight codes. Electronic Journal of Combinatorics, vol. 9 (2002), R41, 23pp.
71. V. S. Pless, W. C. Huffman and R. A. Brualdi. An introduction to algebraic codes. In Handbook of Coding Theory (editors V. S. Pless and W. C. Huffman), North-Holland, 1998, pp. 3-139.
72. N. V. Semakov and V. A. Zinoviev. Balanced codes and tactical configurations. Problems Inform. Transmission, vol. 5 (1969), 22-28.

73. M. Svanström, P. R. J. Östergärd, and G. T. Bogdanova. Bounds and constructions for ternary constant-composition codes. IEEE Trans. Inform. Theory, vol. 48 (2002), 101-111.
74. D. C. Tulpan, H. H. Hoos, and A. E. Condon. Stochastic local search algorithms for DNA word design. In DNA Computing: 8th International Workshop on DNA-Based Computers (editors M. Hagiya and A. Ohuchi), Springer LNCS vol. 2568, 2002, pp. 229-241.
75. D. C. Tulpan and H. H. Hoos. Hybrid randomised neighbourhoods improve stochastic local search for DNA Code design. In Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence (editors Y. Xiang and B. Chaib-draa), Springer LNCS vol. 2671, 2003, pp. 418-433.
76. R. J. M. Vaessens, E. H. L. Aarts, and J. H. van Lint. Genetic algorithms in coding
77. Alf-Steinberger C. The genetic code and error transmission. Proc. Natl. Acad. Sci. USA., 64, p 584-591,.1969 .
78. Dubreil P., Dubreil-Jacotin M. L. Lecciones de álgebra moderna. Editorial Reverté (1963).
79. Friedman S.M. Weinstein I..B.: Lack of fidelity in the translation of ribopolynucleotides. Proc. Natl. Acad. Sci. USA, 52, p 988-996, 1964.
80. Lewin B. Genes VIII. Oxford University Press. 2004.
81. Parker J. 1989. Errors and alternatives in reading the universal genetic code. Microbiol. Rev., 53, 273-298.
82. Redéi L. Algebra, Vol.1. Akadémiai Kiadó., Budapest. 1967.
83. Sánchez, R., Grau, R. and Morgado, E. The Genetic Code Boolean Lattice. Match Commun. Math. Comput. Chem., 52, 29-46, 2004.
84. Sánchez, R., Grau, R., Morgado, E. Grau. A genetic code Boolean structure. I. The meaning of Boolean deductions. Bulletin of Mathematical Biology, 67 p 1–14, 2005.
85. Sánchez, R., Grau, R., Morgado, E. A New DNA Sequences Vector Space on a Genetic Code Galois Field. MATCH Commun. Math. Comput. Chem. 54 (2005), 1, pp: 3-28, 2005.
86. Sánchez, R., Grau, R., Morgado, E. 2005. Gene algebra from a genetic code algebraic structure. J. Math. Biol. Doi: 10.1007/s00285-005-0332-8. WOESE, C.R. On the evolution of the genetic code. Proc. Natl. Acad. Sci. USA., 54, p 1546-1552, 1985

Ярослав Миколайович Николайчук – завідувач кафедри Спеціалізованих комп'ютерних систем факультету Комп'ютерних інформаційних технологій Тернопільського національного економічного університету, доктор технічних наук, професор, академік Української академії наук Національного Прогресу, директор Карпатського державного центру інформаційних засобів і технологій Технічного центру НАН України, член президії асоціації "Вчені Прикарпаття", асоційований член Американського інституту IEEE.